AD-A089 497 ROME AIR DEVELOPMENT CENTER GRIFFISS AFB NY F/G 17/7
HECTIC - A LORAN GRID COMPUTATION PROGRAM FOR IRREGULAR TERRAIN--ETC(U)
FEB 80 J L HECKSCHER, E J TICHOVOLSKY
UNCLASSIFIED RADC-TR-79-354 NL

1 OF 1
AD
A089497

END
DATE
FILMED
1-10-80
DTIC

LEVEL II

(12)

RADC-TR-79-354
In-House Report
February 1980

HECTIC - A LORAN GRID COMPUTATION PROGRAM FOR IRREGULAR TERRAIN

AD A089457

JOHN L. HECKSCHER
ELI J. TICHOVOLSKY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-354 has been reviewed and is approved for publication.

APPROVED: Edward A. Lewis

EDWARD A. LEWIS
Chief, Propagation Branch
Electromagnetic Sciences Division

APPROVED: Allan C. Schell

ALLAN C. SCHELL, Chief
Electromagnetic Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (EEP), Hanscom AFB MA 01731. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER 14 RADC-TR-79-354	2. GOVT ACCESSION NO. AD-A089497
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) 6 HECTIC — A LORAN GRID COMPUTATION PROGRAM FOR IRREGULAR TERRAIN	5. TYPE OF REPORT & PERIOD COVERED In-house
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 John L. Heckscher Eli J. Tichovolsky	8. CONTRACT OR GRANT NUMBER(s) 17 ØØ
9. PERFORMING ORGANIZATION NAME AND ADDRESS Deputy for Electronic Technology (RADC/EEP) Hanscom AFB Massachusetts 01731	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64728F 16 450A0001
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Electronic Technology (RADC/EEP) Hanscom AFB Massachusetts 01731	12. REPORT DATE 11 Feb [redacted] 80
	13. NUMBER OF PAGES 50
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Deputy for Surveillance & Nav Sys (ESD/OCND) Hanscom AFB Massachusetts 01731 12 59	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Arcon Corp.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

LORAN grid prediction
Hufford's integral equation
Propagation over irregular ground
LF navigation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report provides documentation for the computer program HECTIC which calculates LORAN time-difference coordinates over rough terrain. Basic equations are given and the various terms are defined. The program system flow is outlined, and a User's Guide describes the input card, database tape, and printed output formats. A complete listing of the program is given.

DD FORM 1 JAN 73 1473

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

309050 JM

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Preface

The encouragement and support of Dr. E. A. Lewis and the assistance and advice of Mr. W. R. Rustenburg (ESD/OCND) are gratefully acknowledged.

This work was funded in part by the Tactical LORAN System Project Office.

Contents

Illustrations

Tables

HECTIC—A LORAN Grid Computation Program for Irregular Terrain

1. INTRODUCTION

Program HECTIC is a software system for computing time-difference coordinates (TD1 and TD2) produced at specified geographic coordinates by a low frequency LORAN triad. The program, an updated version of HUFLOC,[1] utilizes algorithms originally developed by Johler and Berry[2] and later implemented by Horowitz in LORANCO[3] to compute secondary phase correction factors via Hufford's integral equation.[4] A data base is required which specifies the earth impedance and terrain elevation at points within the geographical region of interest. As implemented on a CDC 6600 computer, HECTIC incorporates the following modifications to the LORANCO system configuration:

(Received for publication 29 January 1980)

1. HUFLOC (1977) Computer Program from USAF, Tactical LORAN SPO, Hanscom AFB, MA 01731.
2. Johler, J. R., and Berry, L. A. (1967) Loran-D Phase Corrections Over Inhomogeneous, Irregular Terrain, ESSA Technical Report IER 59-ITSA 56, (Suptd. of Documents, U.S. Govt Printing Office, Washington, DC 20402).
3. Horowitz, S. (1977) User's Guide for ESD LORAN Grid Prediction Program, RADC-TR-77-407.
4. Hufford, G. A. (1952) An integral equation approach to the problem of wave propagation over an irregular surface, Quart. Appl. Math. 9(No. 4):391.

(1) Output has been limited to data necessary for comparing time-difference predictions with measurements,

(2) New error messages and error recoveries have been added,

(3) Database handler code no longer is hard-wired for a specific geographic area,

(4) Targets below the earth's surface are rejected,

(5) The input card stream was revised, and

(6) The use of blank and labelled COMMON was made more economical.

2. COMPUTER CODE

2.1 Time-of-arrival Function TOA

LORAN time-difference coordinates TD1 and TD2 for a prediction grid serviced by a 3-station chain are given by

$$TD1 = TOA1 - TOAM$$

$$TD2 = TOA2 - TOAM$$

where TOA1 and TOA2 and TOAM are the signal time-of-arrival for Slave 1, Slave 2, and Master stations, respectively. *Time-of-arrival is determined in the* main program HECTIC via

$$TOA = \frac{\eta x}{c} + \frac{10^6 \varphi_c}{\omega} + ED \quad (\mu\,\text{sec})$$

where

η = atmospheric index of refraction at the surface (= 1.000338),

x = geodesic distance from transmitter to target's projection onto a spheroidal earth of equatorial radius a_o and polar radius b_o (m),

c = vacuum velocity of light (= 2.997925×10^6 m/μsec),

ω = radian frequency (= $2\pi f = 2\pi \times 10^5$ Hz),

ED = emission delay (μsec),

and

φ_c = secondary phase correction factor,
= $-\arg(FW) + k(r_o - x)$,

where

F = induction field function FZ or FH,

W = secondary wave attenuation factor,

k = wave number in air ($\omega\eta/c$ m^{-1}),

and

r_o = chordal distance (m) from transmitter to target's projection onto a spherical earth of effective radius a/α where $a = 6.36739 \times 10^6$ (m) and α is the atmospheric vertical lapse factor.

2.2 Induction Field Function F

The induction field is formulated in subroutine INDF as either

$$FZ = \frac{a\, a_r \sin^2\theta}{D^2} + \left[\frac{1}{ikD} + \left(\frac{1}{ikD}\right)^2\right]\left[\frac{3a\, a_r \sin^2\theta}{D^2} - 2\cos\theta\right]$$

or

$$FH = \frac{a_r \sin\theta}{Z_o D}\left[1 + \frac{1}{ikD}\right]$$

where

a_r = $a + h_r$,

h_r = target altitude (m),

D^2 = $a_r^2 + a^2 - 2a\, a_r \cos\theta$,

θ = x/a,*

i = $\sqrt{-1}$, and

Z_o = $\sqrt{\mu_o/\epsilon_o} \simeq 377\ \Omega$.

FZ is used when the phase of the vertical electric field is to be found, and FH is used when the phase of the azimuthal magnetic field is required.

*Although x is the geodesic distance on a spheroidal earth, θ is calculated as if x were the distance on a spherical earth.

2.3 Secondary Wave Attenuation Function W, Surface Case

When the target is on the earth's surface, the quantity W is calculated in subroutine INFQ2F from the following integral equation:

$$W(x) = 1 - e^{i\pi/4}\sqrt{\frac{k}{2\pi}}\;\frac{r_o}{x}\int_o^x ds \cdot W(s)\sqrt{\frac{x}{s\,(x-s)}}$$

$$\cdot \left\{ \delta_s \left[1 + \left(\frac{\partial h_s}{\partial s}\right)^2\right]^{-1/2} + \left[1 + \frac{1}{ik\,r_2}\right]\frac{\partial r_2}{\partial n} \right\}$$

$$\cdot \frac{s(x-s)}{r_1\,r_2}\;\frac{r_o}{x}\left[1 + \left(\frac{\partial h_s}{\partial s}\right)^2\right]^{1/2} e^{-ikr}$$

where

s = distance along the transmitter-to-target geodesic ($0 \le s \le x$),

δ_s = normalized impedance at point s,

= $(k/k_2)\,(1 - k^2/k_2^{\,2})^{1/2}$,

k/k_2 = $(\epsilon_2 - i\sigma/\omega\epsilon_o)^{-1/2}$,

ϵ_2 = relative dielectric constant,

ϵ_o = 8.854×10^{-12} F/m,

σ = conductivity (mho/m),

$\frac{\partial h_s}{\partial s}$ = terrain slope at s,

r = $r_1 + r_2 - r_o$,

$r_1^{\,2}$ = $a_s^{\,2} + a^2 - 2a\,a_s \cos(s/a)$,

a_s = $a + h_s$,

h_s = terrain elevation at point s relative to terrain elevation at the transmitter,

$r_2^{\,2}$ = $a_x^{\,2} + a_s^{\,2} - 2a_x a_s \cos\left(\frac{x-s}{a}\right)$,

a_x = $a + h_x$,

h_x = terrain elevation at point x relative to terrain elevation at the transmitter,

$r_o^{\,2}$ = $a_x^{\,2} + a^2 - 2a_x a \cos\left(\frac{x}{a}\right)$,

$\frac{\partial r_2}{\partial n}$ = normal derivative of r_2 with respect to the earth's surface.

$$= \frac{1}{r_2}\left[1+\left(\frac{a}{a_s}\ \frac{\partial h_s}{\partial s}\right)^2\right]^{-1/2}\left[a_s - a_x \cos\left(\frac{x-s}{a}\right) - a\ \frac{\partial h_s}{\partial s}\ \frac{a_x}{a_s}\sin\left(\frac{x-s}{a}\right)\right].$$

2.4 Secondary Wave Attenuation Function W, Elevated Case

If the target is above the earth's surface, then subroutine INEQ2E also calculates

$$W = \frac{1}{2} - \frac{e^{i\pi/4}}{2}\sqrt{\frac{k}{2\pi}}\ \frac{r_o}{x}\int_0^x ds \cdot W(s)\sqrt{\frac{x}{s\,(x-s)}}$$

$$\cdot\left\{\delta_s\left[1+\left(\frac{\partial h_s}{\partial s}\right)^2\right]^{-1/2} + \left[1+\frac{1}{ik\,r_2}\right]\frac{\partial r_2}{\partial n}\right\}$$

$$\cdot\ \frac{s\,(x-s)}{r_1\,r_2}\ \frac{r_o}{x}\left[1+\left(\frac{\partial h_s}{\partial s}\right)^2\right]^{1/2} e^{-ik\,r}$$

where W(s) is the previously obtained surface function W(x), and the quantity a_x which enters into calculating r_1, r_2, r_o, r and $\partial r_2/\partial_n$ is replaced by $a_r = a + h_r$, where h_r is the target altitude relative to terrain elevation at the transmitter.

3. SYSTEM DESIGN

Program HECTIC is divided into 3 modules: Driver, Database, Handler and Secondary Phase Calculator.

3.1 Driver Module

The Driver routine HECTIC reads data from input cards, checks their validity and converts them from user to internal format via routines CORDMS and CORRAD. It sets up the three geodesic propagation paths from the stations to each target via routines GEODI and GEOPTS. It retrieves impedance and terrain data from the database via the Database Handler routine GEORET. It also calls the Secondary Phase Calculator routine INEQ2E and prints the final results for TD1 and TD2.

3.2 Database Handler Module

The Database Handler routine GEORET reads the database tape and transfers the impedance and terrain data to mass storage via routine GETELV. It also retrieves these data from mass storage via routines GETELV, INDEX, INT, and UNPACK.

3.3 Secondary Phase Calculator Module

The secondary Phase Calculator routine INEQ2E performs initial flat-earth calculations via routines FLEAF and WERF. It interpolates the impedance and terrain data via routines CNEVKEN, GROUND, INT, and OMCOS. Then it evaluates Hufford's integral equation and computes the secondary phase via routine CANG and INDF. These operations are diagrammed in the System Flow Chart in Figure 1.

4. USER'S GUIDE

Program HECTIC reads input cards and a database tape. It produces printer output. These data sets are described below.

4.1 Input Cards

The input file consists of Control, Map, Station, and Target cards. Figure 2 shows a sample input card sequence.

4.1.1 CONTROL CARDS

Cards numbered 1 through 11 in Table 1 are program control cards. These cards specify f, η, a, a_o, b_o, c and α which were defined in Section 2. In addition, the flat-earth distance F_e (~ 1000 m) specifies how far along the path, flat earth theory is to be used to initialize W(s) before round earth code begins. There are also two control switches: one selects either the electric field or the magnetic field; the other allows the selection of either a rough or a smooth, inhomogeneous earth model. (In the smooth model terrain variations are ignored.)

4.1.2 MAP CARDS

The database tape format does not include geographic coordinates because HECTIC automatically keys location to data via the database sequence structure (to be described in Section 4.2). In order to implement this keying operation, the user supplies the location of the first data point via Card 12, and the location 30" east and 30" north of the last data point via Card 13. These locations are referred to as the southwest and northeast map corners, respectively. The input

card format for all geographic positions is ddd-mm-ss.sss A, where ddd = 0 to 90° latitude, or 0 to 180° longitude, mm = 0 to 59 min, ss.sss = 00.000 to 59.999 sec, and A = N, S, E or W. The map corners must lie on integral latitude and longitude lines:

dd-00-00.000 N or S

ddd-00-00.000 E or W

with the following restrictions: (1) the northeast corner cannot lie north of 89°N, (2) the southwest corner cannot lie south of 89°S, (3) the map cannot straddle the 180° meridian, and (4) the minimum map region is a zone 1° latitude by 2° longitude, with larger map regions being multiples of this basic zone.

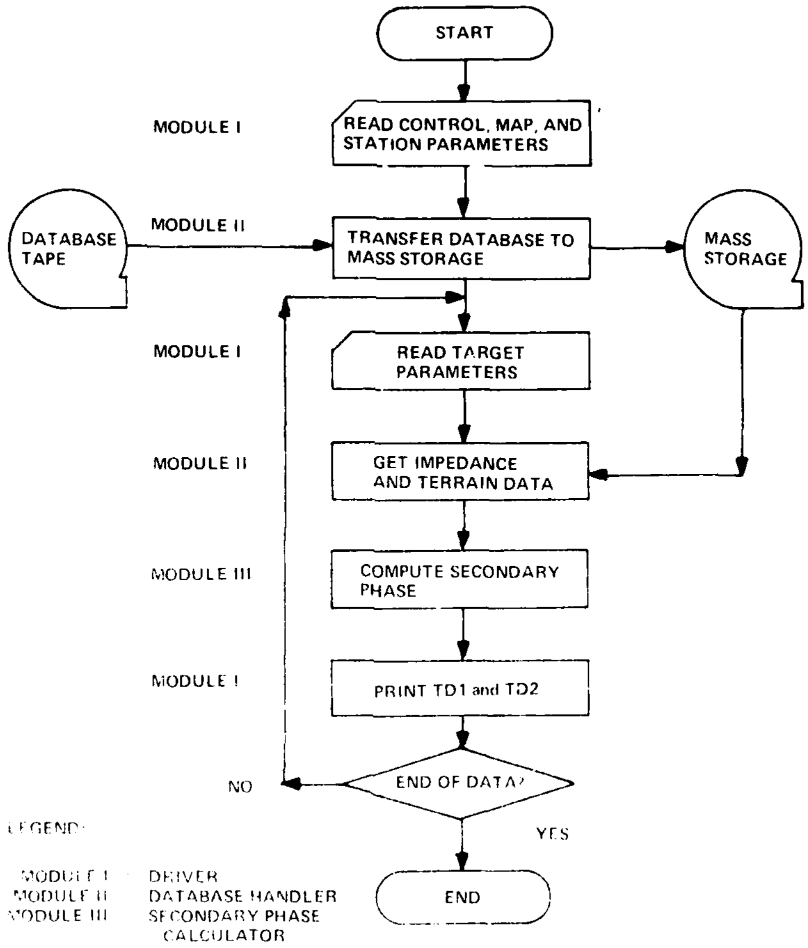

Figure 1. System Flow Chart

```
  1 100.        .FREQUENCY (KHZ)
  2 1.000338 .INDEX OF REFRACTION OF AIR AT GROUND LEVEL
  3 6367390. .SPHERICAL EARTH RADIUS (M)
  4 6378388. .SPHEROIDAL EARTH EQUATORIAL RADIUS (M)
  5 6356911.9.SPHEROIDAL EARTH POLAR RADIUS (M)
  9 .2997925 .VACUUM SPEED OF LIGHT (KM/USEC)
 11 1000.       .FLAT EARTH TEST FACTOR
 12 .85         .TRUE RADIUS/EFFECTIVE RADIUS
 14 1.          .ZERO=H-FIELD, NON-ZERO=E-FIELD
 15 0.          .UNITY=SMOOTH EARTH, NON-UNITY=ROUGH EARTH.
                .CONTROL CARDS TERMINATOR. MAP, STATION, AND TARGET CARDS FOLLOW.
MAP S.W.        46 00 00.000N     6 00 00.000E
MAP N.E.        54 00 00.000N    14 00 00.000E
MASTER          49 36 18.813N    07 19 38.276E
SLAVE 1         53 34 13.867N    08 43 46.508E 12537.42
SLAVE 2         48 15 48.929N    11 37 44.263E 26164.17
TARGET    1     49 45 27.518N     9  2 47.537E   3048.00        10.
TARGET    2     49 44 52.655N     9 14 11.276E   3048.00        10.
TARGET    3     49 44 23.195N     9 15 25.244E   3048.00        10.
TARGET    4     49 39    .565N    9 26 46.355E   3048.00        10.
TARGET    5     49 36 33.185N     9 29 54.484E   3048.00        10.
TARGET    6     49 33 39.426N     9 42  7.803E                  10.
TARGET    7     49 33  8.219N     9 43 24.594E                  10.
TARGET    8     49 29 17.213N     9 53  1.464E                  10.
TARGET    9     49 28 50.696N     9 54  4.958E                  10.
TARGET   10     49 26  6.593N     9  2 22.280E                  10.
TERMINATOR
```

Figure 2. Sample Input Card Images

Table 1. Input Card Formats

Card Number	Card Column(s)	Read Format	Data Entry
1	5 6-16	I5 E10.0	1 Loran transmission frequency f (kHz)
2	5 6-16	I5 E10.0	2 Air index of refraction at ground η
3	5 6-16	I5 E10.0	3 Spherical earth radius a (m)
4	5 6-16	I5 E10.0	4 Spheroidal earth equatorial radius a_o (m)
5	5 6-16	I5 E10.0	5 Spheroidal earth polar radius b_o (m)
6	5 6-16	I5 E10.0	9 Vacuum speed of light c (km/μsec)
7	4-5 6-16	I5 E10.0	11 Flat-earth distance F_e (m)
8	4-5 6-16	I5 E10.0	12 Vertical lapse factor α
9	4-5 6-16	I5 E10.0	14 1.0 for electric field prediction; 0.0 for magnetic field prediction.

Table 1. Input Card Formats (Cont)

Card Number	Card Column(s)	Read Format	Data Entry
10	4-5	I5	15
	6-16	F10.0	0.0 for data base rough earth terrain; 1.0 for smooth earth.
11	1-5		Blank. Terminates Control Card stream.
12	2-12	A10	MAP SW.
	13-17	I5	Map SW corner latitude degrees.
	18-20	I3	Map SW corner latitude minutes.
	21-27	F7.3	MAP SW corner latitude seconds.
	28	A1	Map SW corner hemisphere, N or S.
	29-33	I5	Map SW corner longitude degrees.
	34-36	I3	Map SW corner longitude minutes.
	37-43	F7.3	Map SW corner longitude seconds.
	44	A1	Map SW corner hemisphere, F or W.
13	2-12	A10	MAP NE.
	13-44		Map NE corner. Use the formats of Card 12.
14	2-12	A10	MASTER.
	13-44		Master Station position. Use the formats of Card 12.
15	2-12	A10	SLAVE 1.
	13-44		Slave 1 position. Use the formats of Card 12.
	45-53	F9.2	Slave 1 emission delay (μsec).
16	2-12	A10	SLAVE 2.
	13-44		Slave 2 position. Use the formats of Card 12.
	45-53	F9.2	Slave 2 emission delay (μsec)
17	2-12	A10	Target identifier.
	13-44		Target position. Use the formats of Card 12.
	45-53	F9.2	Target altitude h_r (m)
	54-62	F9.2	Hufford's integral equation path increment Δs (m)
	63-71	F9.2	Measured TD1 (μsec)
	72-80	F9.2	Measured TD2 (μsec)
18, etc.	2-80		Next Target Cards. Use formats of Card 17.
LAST CARD	54-62		Blank. Terminates the Input Card Stream.

4.1.3 STATION CARDS

Cards 14, 15, and 16 specify the positions of the Master, Slave 1, and Slave 2, respectively. Cards 15 and 16 also contain the emission delays for the Slaves.

4.1.4 TARGET CARDS

Card 17 contains target identifier, position, altitude h_r, integration path increment size Δs, and measured TD1 and TD2. The latter pair may be omitted.

Any number of additional targets may be specified, one to a card. Any target card with $\Delta \leqq 0$ will prematurely stop the program. To properly end the program, a blank card should be inserted after the last target card.

4.2 Database Tape

Program HFCTIC reads terrain elevation, impedance amplitude, and impedance phase from the database tape. The tape contains one such "data triad" for every 30 sec of latitude and longitude. Each triad is packed into 30 bits, and since the word size available on a CDC 6600 computer is 60 bits, two data triads fit into each 60-bit word. The right half-word always contains data for the geographic point 30-sec north of the point in the left half-word, as indicated in Table 2.

One database record contains sixty 60-bit words, for a total of 120 data triads. A record thus spans 1° of latitude from south to north, as shown in Figure 3(a). Two-hundred-forty such records fill a database zone which spans 2° of longitude from west to east in 30-sec steps.

The database tape must contain enough zones to completely cover the region specified via the map corner coordinates. An example of the ordering of zones on the database tape is diagrammed in Figure 3(b). Data pertinent to zone 1 appears in records 1 through 240, to zone 2 in records 241 through 480, and so on.

Table 2. Database Word Packing

Half	Bit Numbers	Data Description	Least Significant Bit
Left	59-49	Terrain Elevation	8 m
	48-41	Impedance Amplitude	0.001 Ω
	40-30	Impedance Phase	0.001 rad
Right	29-19 18-11 10-0	(Repeat above sequence for geographic point 30 sec north)	(as above)

4.3 Printed Output

The printed output consists of Normal Output and Error Messages. Normal Output includes Control, Map, and Station parameters, TD1 and TD2 predictions, and the End Message. Figure 4 shows a sample normal output listing. Error messages may appear in this listing in the event of either fatal or nonfatal errors.

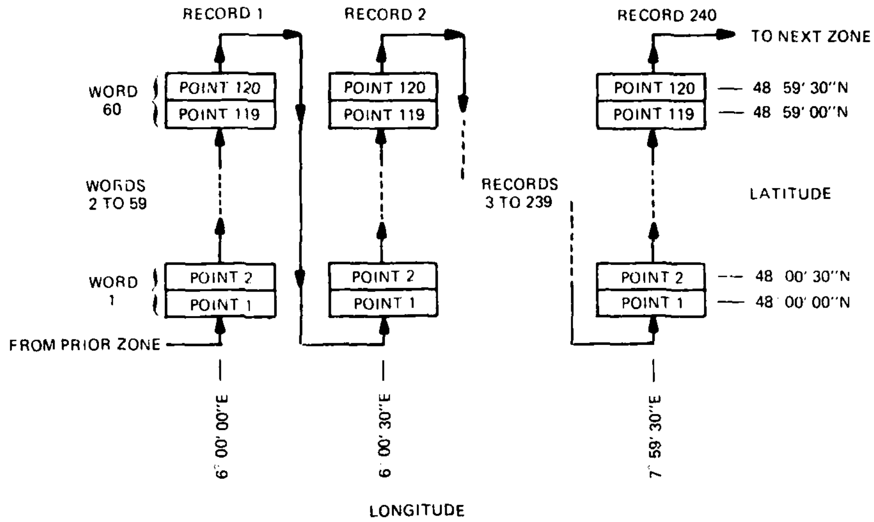

Figure 3(a). Sample Database Zone Structure, Showing Ordered Sequence of Database Words and Corresponding Geographic Coordinates

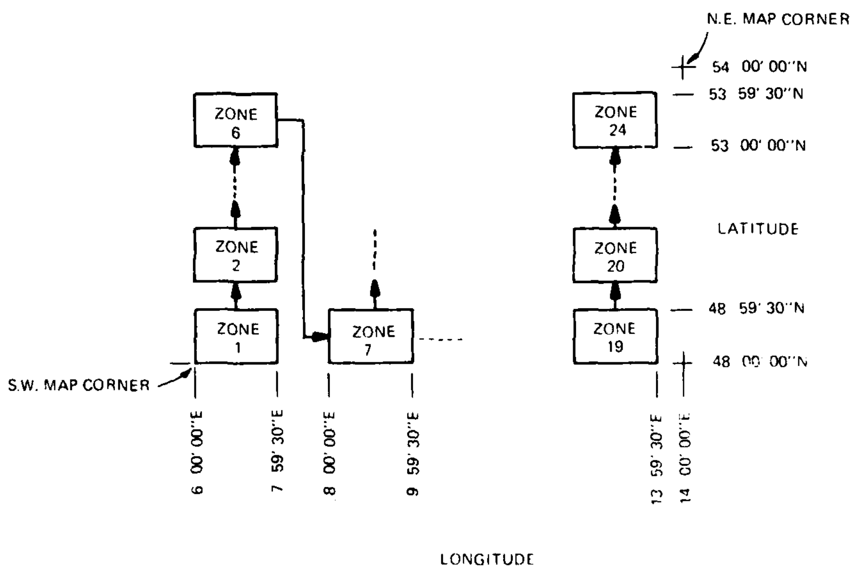

Figure 3(b). Sample Map Structure, Showing Ordered Sequence of Zones and Corresponding Geographic Coordinates. The coordinates of the southwest and northeast map corners determine the number of zones in each direction

```
- - - - - - - - - - - - - - -CONTROL PARAMETERS- - - - - - - - - - - - - -
.1[illegible]0000000000000E+03    ARRAY(1) = FRMZ = FREQUENCY (MHZ)
.1[illegible]00338000000000E+01   ARRAY(2) = ETA = AIR INDEX OF REFRACTION AT GROUND
.6367390000000000E+07   ARRAY(3) = SPHRAD = SPHERICAL EARTH RADIUS (M)
.6378388000000000E+07   ARRAY(4) = A0 = SPHEROIDAL EARTH EQUATORIAL RADIUS (M)
.6356911900000000E+07   ARRAY(5) = B0 = SPHEROIDAL EARTH POLAR RADIUS (M)
[illegible]             ARRAY(6) = FE = SPHEROIDAL FLATTENING
[illegible]             ARRAY(7) = ES = SPHEROIDAL FIRST ECCENTRICITY SQUARED
[illegible]
```

- - - - - - - - -MAP PARAMETERS- - - - - - - -

IDENTIFIER	LATITUDE (DEG-MIN-SEC)	LONGITUDE (DEG-MIN-SEC)
MAP S.W.	48 [illegible]	[illegible] 0.0 [illegible]
MAP N.E.	[illegible]	14 0 0.0 [illegible]

- - - - - - - - - - -STATION PARAMETERS- - - - - - - - - -

IDENTIFIER	LATITUDE (DEG-MIN-SEC)	LONGITUDE (DEG-MIN-SEC)	EMISSION DELAY (USEC)
MASTER	49 3[illegible] 1[illegible].[illegible]13N	7 19 36.[illegible]7[illegible]E	[illegible].00
SLAVE 1	53 39 13.[illegible]7N	[illegible] 43 46.50[illegible]E	125[illegible]7.[illegible]2
SLAVE 2	48 1[illegible] [illegible]	11 37 49.2[illegible]E	26166.17

(a)

- -LORAN TIME DIFFERENCE PREDICTIONS- - - - - - - - - - - - - - - - - - -

IDENTIFIER	LATITUDE (DEG-MIN-SEC)	LONGITUDE (DEG-MIN-SEC)	ALTITUDE (METER)	STEPSIZE (KM)	--MEASURED(USEC)-- TD1	TD2	--COMPUTED(USEC)-- TD1	TD2	---ERROR(USEC)--- TD1	TD2
TARGET 1	49 44 27.516N	9 2 47.637E	3048.00	10.00	0.00	0.00	13540.49	26598.87	13540.49	26598.87
TARGET 2	49 44 52.665N	9 14 11.276E	3048.00	10.00	0.00	0.00	13530.47	26504.72	13530.47	26504.72
TARGET 3	49 44 23.195N	9 15 25.246E	3048.00	10.00	0.00	0.00	13529.23	26494.36	13529.23	26494.36
TARGET 4	49 39 .565N	9 28 46.358E	3048.00	10.00	0.00	0.00	13515.71	26380.68	13515.71	26380.68
TARGET 5	49 38 33.185N	9 29 54.489E	3048.00	10.00	0.00	0.00	13514.57	26370.93	13514.57	26370.93
TARGET 6	49 33 39.426N	9 42 7.803E	0.00	10.00	0.00	0.00	13501.30	26264.93	13501.30	26264.93
TARGET 7	49 33 8.621N	9 43 24.569E	0.00	10.00	0.00	0.00	13499.83	26253.69	13499.83	26253.69
TARGET 8	49 29 17.213N	9 53 1.989E	0.00	10.00	0.00	0.00	13489.05	26169.26	13489.05	26169.26
TARGET 9	49 28 50.896N	9 54 4.958E	0.00	10.00	0.00	0.00	13488.00	26159.87	13488.00	26159.87
TARGET 10	49 26 6.693N	9 2 22.280E	0.00	10.00	0.00	0.00	13688.19	26515.16	13688.19	26515.16
TERMINATOR	0 0 0.000	0 0 0.000	0.00	0.00	0.00	0.00				

END OF INPUT DATA.

(b)

Figure 4. Sample Normal Output, (a) Control, map, and station parameters corresponding to input sequence of Figure 2. (b) LORAN time-difference predictions for target cards of Figure 2

4.3.1 CONTROL PARAMETERS

The control parameters are f, η, a, a_o, b_o, c, F_e, α, electric/magnetic field control switch, and rough/smooth earth switch. These were defined in Sections 2 and 3. Also included in this listing are the calculated quantities k (wavenumber), a/α (effective earth radius), and the spheroidal flattening

$$f_e = 1 - a_o/b_o$$

and the first spheroidal eccentricity squared

$$e^2 = 1 - a_o^2/b_o^2 .$$

4.3.2 MAP PARAMETERS

The Map parameters are the southwest and northeast map corners defined in Section 4.1.2.

4.3.3 STATIONS PARAMETERS

The station parameters are the positions and emissions delays of the Master, Slave 1, and Slave transmitters defined in Section 5.1.3.

4.3.4 TD1 AND TD2 PREDICTIONS

For each target there is one printed line which includes identifier, latitude, longitude, altitude, numerical integration step size, and measured TD1 and TD2 defined in Section 5.1.4. Also included are the computed TD1 and TD2 and the time delay errors (computed minus measured) in microseconds.

4.3.5 END MESSAGE

The message "End of Input Data" indicates normal run completion.

4.3.6 ERROR MESSAGES

A fatal error is one from which there is no reasonable recovery. Hence, the program must abort. The user merely corrects the fault described below and reruns the program. On the other hand, for a nonfatal error the code either takes remedial action or it abandons the current target and advances to the next one. Error messages are grouped below under the routine which generates them. The message number refers to the FORTRAN statement number of the message.

4.3.6.1 Program HECTIC Error Messages

Message 9900 (Fatal)

"...Station lies outside map area. Run aborts." Fault: Input station coordinates exceeded the boundaries specified on the map input cords. All three transmitters must lie within the map region.

Message 9910 (Fatal)

"Error in data for ... coordinates. Run ends." Fault: Map or station coordinates on input card were not correct. Either the specified latitude degrees exceeded 90, longitude degrees exceed 180, minutes exceeded 59, seconds exceeded 59.999, latitude symbol was neither N nor S, or longitude symbol was neither E nor W.

Message 9912 (Nonfatal)

"Error in end-point coordinate. Calculations for this path have been deleted." Fault: Target coordinates on input card were not correct. For diagnosis, see

message 9910 above. Action: The program advances to the next target.

Message 9916 (Nonfatal)

"Error detected during conversion of coordinate for point—from radians to (alphameric) degrees-minutes-seconds." Fault: A point on a station-to-target path was not correct. Either latitude exceeded 324000. 0005 seconds, or longitude exceeded 648000. 0005 seconds. Action: The program advances to the next target.

Message 9922 (Nonfatal)

"Target lies outside map area. Get next target." Fault: Input target coordinate exceeded the boundaries specified on the map input cards. Action: The program advances to the next target.

4.3.6.2 Subroutine GEOPTS Error Messages

Message 9800 (Fatal)

"Subroutine GEOPTS called with end-point latitude out of acceptable range $(-\pi/2, \pi/2)$. Latitude of A = ... radians, latitude of B = ... radians. Program execution terminated." Fault: GEOPTS encountered a bad latitude.

Message 9802 (Fatal)

"Subroutine GEOPTS called with end-point longitude out of acceptable range $(-2\pi, +2\pi)$. Longitude of A = ... radians, longitude of B = ... radians. Program execution terminated." Fault: GEOPTS encountered a bad longitude.

Message 9803 (Fatal)

"Geodesic path includes a geographic pole—subroutine GEOPTS cannot handle this case."

4.3.6.3 Subroutine GETELV Error Messages

Message 9700 (Fatal)

"Database has nnn records. Increase array MSINDX and variable INDDIM to accommodate. Run Aborts." Fault: Subroutine GETELV computed the number of records which the database tape must contain in order to be compatible with the specified map region. This number exceeded the number of mass-storage record pointer locations available. Consequently, the database could not be loaded. Remedy: the user should increase the size of array MSINDX to nnn + 1 and reset the value of INDDIM to nnn + 1. Array MSINDX is in blank common and INDDIM is in a data statement in routine GETELV.

Message 9701 (Nonfatal)

"Lat, lon requested are outside map region — no elevation returned." Fault: The program automatically increments any given path by two stepsizes beyond the target location. Consequently, the path for a target that initially was within the map region ended up crossing a map boundary. Action: The program proceeds, but the results for this target are questionable. The user may try a smaller step size for this target.

Message 9702 (Nonfatal)

"Index requested exceeds scanlines generated, thus previous elevation will be used." Fault: The record pointer returned by subroutine INDEX to GETELV exceeded the number of records on the database tape. Action: The program proceeds, but the results for this target are questionable.

4.3.6.4 Subroutine CNEVKEN Error Messages

Message 9602 (Fatal)

"There are not enough points in the given array." Fault: CNEVKEN tried to interpolate a function which was defined over an insufficient range of values.

Message 9600 (Fatal)

"The X values are not arranged in ascending order.

I = . . . , X(I) = . . . , J = . . . , A(J) = . . .

X F(X)

. .

. .

. .

Fault: CNEVKEN interpolates the funxtion F(X) over array X from a given function F(A) over array A. Both X and A must be in ascending order. The X array was not in order.

4.3.6.5 Subroutine INDF Error Messages

Message 9500 (Fatal)

"In INDF, distance is zero or negative, XX = . . ." Fault: INDF tried to calculate the induction field for a meaningless distance stored in variable XX.

5. PROGRAM LISTING

```
*MODULE 1.  4/22/80
*DECK HECTIC
      PROGRAM HECTIC(INPUT=128,OUTPUT=128,TAPE13=128,TAPE2=128)
C INSTRUCTIONS FOR USE:-------------------
C  ATTACH,IN,INPUTCARDS,ID=TICHO,MR=1.
C  ATTACH,HECTIC,ID=TICHO,MR=1.
C  FTN,I=HECTIC,EL=F,OPT=1,L=0.
C  VSN,TAPE13=CC3626.
C  LABEL,TAPE13,R,NORING,L=DATABASE.
C  LGO,IN.
C END OF INSTRUCTIONS.
C PROGRAM DESCRIPTION:-------------------
C MODULE 1 READS INPUT CARDS, CALL MODULES 2 AND 3, AND PRINTS
C FINAL RESULTS.
C  ROUTINES: CORDMS,CORRAD,GEODI,HECTIC.
C MODULE 2 GETS ALTITUDES AND IMPEDANCES FOR A STATION-TO-TARGET
C PATH. READS DATABASE TAPE IN FIRST PASS ONLY.
C  ROUTINES: GEOPTS,PCOORD,GEORET,GETELV,INDEX,INT,UNPACK.
C MODULE 3 COMPUTES THE SECONDARY PHASE FACTOR.
C  ROUTINES: CANG,CNEVKEN,GEOM,GROUND,FLEAF,INDF,INEQ2E,
C            OMCOS,WERF.
C FILES USED:-------------------
C  INPUT - FOR INPUT CARDS.
C  OUTPUT - FOR RUN STATUS, ERROR MESSAGES AND DEBUGGING.
C  TAPE2 - FOR MASS STORAGE.
C  TAPE13 - FOR DATABASE INPUT TAPE.
C REVISED BY:-------------------
C  ELI TICHOVOLSKY, ARCON CORP., WALTHAM, MA., 4/22/80.
C ORIGINAL BY:-------------------
C  J. R. JOHLER, ET. AL.
C ENTRY POINTS:----------------------
C  PCOORD IS IN GEOPTS, GEOM IS IN GROUND.
C --------------------------
      DIMENSION TPW(3),ED(2),TD(2),IDENTF(4),TDMEAS(2),TDERR(2),TOA(3)
      DIMENSION LDUM(3,4),ADUM(3,4),RLA(3),RLO(3),LATA(3),LONA(3)
      COMMON /ZOTA/ARRAY(15)
      COMMON /PIS/TWOPI,PI,HAFPI,QRTPI
      COMMON /MAP/LATSW,LATNE,LONSW,LONNE
      COMMON /PATH/RLATA,RLONA,RLATB,RLONB,RAZA,RAZB,SBM
      COMMON /CITCT/ITCT,LTOP,NPTCT,ZXMTR,ZMIN,ALTSW(3),TSW(3)
      COMMON LAT(999),LON(999),MSINDX(5800)
      EQUIVALENCE (ARRAY(2),ETA),(ARRAY(11),FLAT)
      EQUIVALENCE (ARRAY(12),ALPHA),(ARRAY(14),EORH),(ARRAY(15),SMOOTH)
      EQUIVALENCE (ARRAY(1),FKHZ),(ARRAY(13),RAD),(ARRAY(3),SPHRAD)
      EQUIVALENCE (ARRAY(4),A0),(ARRAY(5),B0),(ARRAY(9),C)
      EQUIVALENCE (ARRAY(6),FL),(ARRAY(7),ESQ),(ARRAY(10),WAVE)
C SEMI-MAJOR AND SEMI-MINOR AXES (METER) OF THE INTERNATIONAL SPHEROID.
      DATA A0,B0,SPHRAD/6378388.0, 6356911.9461, 6367390./
      DATA LINE,ITCT,PI/50, 1, 3.14159265358 98/
      DATA FKHZ,ETA,C/100.,1.000338,.2997925/
      DATA FLAT,ALPHA,EORH,SMOOTH/1000.,.85,1.,0./
      DATA WEST,SOUTH,LWEST,LSOUTH/1HW,1HS,1HW,1HS/
      TWOPI=2.*PI $HAFPI=PI/2. $QRTPI=PI/4.
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C READ PROCESSING CONTROLS.
      I=0
20    READ 9919,I,AVALUE $IF (I .LE. 0) GO TO 21
```

```
      ARRAY(I)=AVALUE $GO TO 20
21    CONTINUE
C RAD=EFFECTIVE EARTH RADIUS. WAVE=WAVENUMBER IN AIR AT EARTH SURFACE.
      RAD = SPHRAD/ALPHA $WAVE = TWOPI*FRHZ*ETA*1.E-6/C
C FL=SPHEROIDAL FLATTENING. ESQ=FIRST ECCENTRICITY SQUARED.
      FL = 1. - B0/A0 $ESQ = 1. - (B0/A0)**2
      PRINT 9920, (ARRAY(I),I=1,6)
      PRINT 9921, (ARRAY(I),I=7,15)
      PRINT 9925
C READ MAP CORNERS.
      DO 911 I=1,2
      READ 9909,IDENTF(I),LDUM(I,1),LDUM(I,2),ADUM(I,1),ADUM(I,2),
     1          LDUM(I,3),LDUM(I,4),ADUM(I,3),ADUM(I,4)
      PRINT 9909,IDENTF(I),LDUM(I,1),LDUM(I,2),ADUM(I,1),ADUM(I,2),
     1          LDUM(I,3),LDUM(I,4),ADUM(I,3),ADUM(I,4)
      MIS=0
      CALL CORRAD(RLA(I),LDUM(I,1),LDUM(I,2),ADUM(I,1),ADUM(I,2),0,MIS)
      CALL CORRAD(RLO(I),LDUM(I,3),LDUM(I,4),ADUM(I,3),ADUM(I,4),1,MIS)
      IF (MIS .EQ. 0) GO TO 911
C IF MAP COORDINATES ARE BAD, ABORT.
      PRINT 9910, IDENTF(I) $STOP
911   CONTINUE
      LATSW=LDUM(1,1)*10000+LDUM(1,2)*100+IFIX(ADUM(1,1))
      IF (ADUM(1,2) .EQ. SOUTH) LATSW=-LATSW
      LONSW=LDUM(1,3)*10000+LDUM(1,4)*100+IFIX(ADUM(1,3))
      IF (ADUM(1,4) .EQ. WEST) LONSW=-LONSW
      LATNE=LDUM(2,1)*10000+LDUM(2,2)*100+IFIX(ADUM(2,1))
      IF (ADUM(2,2) .EQ. SOUTH) LATNE=-LATNE
      LONNE=LDUM(2,3)*10000+LDUM(2,4)*100+IFIX(ADUM(2,3))
      IF (ADUM(2,4) .EQ. WEST) LONNE=-LONNE
C READ TRANSMITTER COORDINATES FOR MASTER, SLAVE1, AND SLAVE2.
      PRINT 9926
      DO 950 I=1,3
      READ 9909,IDENTF(I),LDUM(I,1),LDUM(I,2),ADUM(I,1),ADUM(I,2),
     1          LDUM(I,3),LDUM(I,4),ADUM(I,3),ADUM(I,4),DATUM
      PRINT 9909,IDENTF(I),LDUM(I,1),LDUM(I,2),ADUM(I,1),ADUM(I,2),
     1          LDUM(I,3),LDUM(I,4),ADUM(I,3),ADUM(I,4),DATUM
      IF (I .GE. 2) ED(I-1) = DATUM
      MIS = 0
      CALL CORRAD(RLO(I),LDUM(I,3),LDUM(I,4),ADUM(I,3),ADUM(I,4),1,MIS)
      CALL CORRAD(RLA(I),LDUM(I,1),LDUM(I,2),ADUM(I,1),ADUM(I,2),0,MIS)
      IF (MIS .EQ. 0) GO TO 940
C IF STATION COORDINATES ARE BAD, ABORT.
      PRINT 9910,IDENTF(I) $STOP
940   LATA(I)=LDUM(I,1)*10000+LDUM(I,2)*100+IFIX(ADUM(I,1))
      IF (ADUM(I,2) .EQ. SOUTH) LATA(I)=-LATA(I)
      LONA(I)=LDUM(I,3)*10000+LDUM(I,4)*100+IFIX(ADUM(I,3))
      IF (ADUM(I,4) .EQ. WEST) LONA(I)=-LONA(I)
      IF (LATA(I).GE.LATSW  .AND.  LATA(I).LT.LATNE  .AND.
     1    LONA(I).GE.LONSW  .AND.  LONA(I).LT.LONNE)  GO TO 950
C IF STATION LIES OUTSIDE MAP AREA, ABORT.
      PRINT 9900,IDENTF(I) $STOP
  950 CONTINUE
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C START LOOP ON TARGETS.
    5 CONTINUE
      IF (LINE .LT. 50) GO TO 55
```

```
      PRINT 9924 $LINE=0
C READ TARGET COORDINATES. ZMIN = AIRBORNE-TARGET ALTITIUDE (M) ABOVE
C MEAN SEA LEVEL IF NON-ZERO. ADELS = PATH INCREMENT SIZE (KM).
55    READ 9909,IDENTF(4),LATDEGB,LATMINB,SECLATB,LATIDB,
     .              LONDEGB,LONMINB,SECLONB,LONIDB,ZMIN,ADELS,TDMEAS
      PRINT 9909,IDENTF(4),LATDEGB,LATMINB,SECLATB,LATIDB,
     1              LONDEGB,LONMINB,SECLONB,LONIDB,ZMIN,ADELS,TDMEAS
      LINE = LINE + 1
C STOP IF PATH INCREMENT SIZE IS ZERO OR LESS.
      IF(ADELS .LE. 0.) GO TO 10
      MIS = 0
      CALL CORRAD(RLATB,LATDEGB,LATMINB,SECLATB,LATIDB,0,MIS)
      CALL CORRAD(RLONB,LONDEGB,LONMINB,SECLONB,LONIDB,1,MIS)
      IF (MIS .EQ. 0) GO TO 15
C IF TARGET COORDINATES ARE BAD, GET ANOTHER TARGET.
      PRINT 9912 $LINE=LINE+3
      GO TO 5
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C START LOOP ON 3 STATION-TO-TARGET PATHS.
   15 CONTINUE
      RLATA=RLA(ITCT)
      RLONA=RLO(ITCT)
      LTOP=0
      ALTSW(ITCT) = TSW(ITCT) = SBKMS = 0.
C SKIP THIS PATH IF IT IS TOO SHORT.
      IF ((ABS(RLATA-RLATB) .LE. ADELS/SPHRAD) .AND.
     1    (ABS(RLONA-RLONB) .LE. ADELS/SPHRAD)) GO TO 16
C COMPUTE SPHEROIDAL BASELINE AND FORWARD AND BACK AZIMUTHS.
      CALL GEOPTS(0.0,RLATP,RLONP,RAZP)
C COMPUTE SPHEROIDAL BASELINE AND FORWARD AND BACK AZIMUTHS.
      CALL GEODI(RLATA,RLONA,RAZAS,SBMS,RLATB,RLONB,RAZBS)
      SBKMS=SBMS*.001
      SBKM = SBM*1.0E-3
      DSKM=NPTCT=SBKM/ADELS+1.0
      DSM = SBM/DSKM
      DSKM = DSM*1.0E-3
C SET TRANSMITTER COORDINATES AS POINT NUMBER 1.
      LAT(1)=LATA(ITCT) $LON(1)=LONA(ITCT)
C ADD 2 POINTS BEYOND RECEIVER.
      NPP2=NPTCT+2
      SPM = 0.0
C COMPUTE LATITUDES AND LONGITUDES ALONG HUFFORD'S BASELINE.
      DO 100 IP=1,NPP2
      SPM = SPM+DSM
      CALL PCOORD(SPM,RLATP,RLONP,RAZP)
      MIS = 0
      CALL CORDMS(RLATP,LATDEGP,LATMINP,SECLATP,LATIDP,0,MIS)
      CALL CORDMS(RLONP,LONDEGP,LONMINP,SECLONP,LONIDP,1,MIS)
      IF (MIS .EQ. 0) GO TO 35
C IF PATH COORDINATE IS BAD, GET ANOTHER TARGET.
      PRINT 9916,IP $LINE=LINE+4
      GO TO 5
35    CONTINUE
      LAT(IP+1)=LATDEGP*10000+LATMINP*100+IFIX(SECLATP)
      IF (LATIDP .EQ. LSOUTH) LAT(IP+1)=-LAT(IP+1)
      LON(IP+1)=LONDEGP*10000+LONMINP*100+IFIX(SECLONP)
      IF (LONIDP .EQ. LWEST) LON(IP+1)=-LON(IP+1)
```

```
  100 CONTINUE
C ADD 1 POINT FOR TRANSMITTER.
      IP=NPTCT+3
C IF TARGET IS OUTSIDE MAP AREA. GET ANOTHER TARGET.
      IF (LAT(IP).GE.LATSW  .AND.  LAT(IP).LT.LATNE  .AND.
     1    LON(IP).GE.LONSW  .AND.  LON(IP).LT.LONNE) GO TO 850
      PRINT 9922,IDENTF(4) $LINE=LINE+1 $GO TO 5
C COMPUTE ALTITUDES AND IMPEDANCES ALONG HUFFORD'S BASELINE.
850   CALL GEORET(DSKM,IP)
C ADJUST NUMBER OF POINTS FOR ROUTINE INEQ2E.
      NPTCT = NPTCT + 1
C COMPUTE HUFFORD'S EQUATION.
      CALL INEQ2E
16    CONTINUE
      TPW(ITCT) = ETA*SBKMS/C
      ITCT=ITCT+1
      IF (ITCT .LT. 4) GO TO 15
C END LOOP ON 3 STATION-TO-TARGET PATHS.
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C PRINT FINAL RESULTS.
      DO 9 I=1,3
      IF (LTOP.EQ.0) TOA(I) = TPW(I) + TSW(I)
      IF (LTOP.EQ.1) TOA(I) = TPW(I) + ALTSW(I)
9     CONTINUE
      DO 710 I =1,2
      TD(I) = ED(I) + TOA(I+1) - TOA(1)
710   TDERR(I) = TD(I) - TDMEAS(I)
      PRINT 9923,TD,TDERR
C GET NEXT TARGET.
      ITCT=1
      GO TO 5
C END LOOP ON TARGETS.
C- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
10    PRINT 9918
      STOP
9900  FORMAT (* *A10*  STATION LIES OUTSIDE MAP AREA. RUN ABORTS.*)
9909  FORMAT(2X,A10,2(I5,I3,F7.3,A1),4F9.2)
9910  FORMAT(*0 ERROR IN DATA FOR *A10*  COORDINATES. RUN ENDS.*)
9912  FORMAT(//11X, 3H***, 9X, 29HERROR IN END-POINT COORDINATE. 10X, 3H
     +***/11X,54H***  CALCULATIONS FOR THIS PATH HAVE BEEN DELETED  ***)
9916  FORMAT(//2X, 62H***  ERROR DETECTED DURING CONVERSION OF COORDINAT
     +ES FOR POINT, I5, 5H  ***/2X, 2H**, 7X, 52HFROM RADIANS TO (ALPHAM
     +ERIC) DEGREES-MINUTES-SECONDS, 7X, 3H***/2X, 3H***, 66X, 3H***/2X,
     + 3H***, 7X, 52HFURTHER CALCULATIONS FOR THIS PATH HAVE BEEN DELETE
     +D, 7X, 3H***)
9918  FORMAT(*0 END OF INPUT DATA.*)
9919  FORMAT(I5,E10.0)
9920  FORMAT  (1H1//15(2H -)*CONTROL PARAMETERS*15(2H- )/
     AE22.14,5X*+ARRAY(1) = FKHZ = FREQUENCY (KHZ)*/
     1E22.14,5X*+ARRAY(2) = ETA = AIR INDEX OF REFRACTION AT GROUND*/
     2E22.14,5X*+ARRAY(3) = SPHRAD = SPHERICAL EARTH RADIUS (M)*/
     4E22.14,5X*+ARRAY(4) = A0 = SPHEROIDAL EARTH EQUATORIAL RADIUS (M)*
     5/E22.14,5X*+ARRAY(5) = H0 = SPHEROIDAL EARTH POLAR RADIUS (M)*/
     6E22.14,5X* ARRAY(6) = FL = SPHEROIDAL FLATTENING*)
9921  FORMAT  (E22.14,5X* ARRAY(7) = ESQ = SPHEROIDAL FIRST ECCENTRICITY
     2 SQUARED*/
     2E22.14,5X* ARRAY(8) = NOT USED*/
```

```
     4E22.14,5X*+ARRAY(9) = VACUUM SPEED OF LIGHT (KM/USEC)*/
     5E22.14,5X* ARRAY(10) = WAVE = WAVENUMBER (1/M) IN AIR AT EARTH SUR
     7FACE*/
     6E22.14,5X*+ARRAY(11) = FLAT = FLAT EARTH FACTOR (M)*/
     8E22.14,5X*+ARRAY(12) = ALPHA = VERTICAL LAPSE FACTOR*/
     9E22.14,5X* ARRAY(13) = EFFECTIVE EARTH RADIUS (M)*/
     1E22.14,5X*+ARRAY(14) = EORH = 0. FOR H-FIELD, NON-0. FOR E-FIELD*/
     9E22.14,5X*+ARRAY(15) = SMOOTH = 1. FOR SMOOTH EARTH, NON-1. FOR RO
     2UGH EARTH*/
     327X*+ ITEMS MARKED WITH + MAY BE CHANGED VIA INPUT CARDS.*/)
9922  FORMAT (* *A10*  TARGET LIES OUTSIDE MAP AREA. GET NEXT TARGET.*)
9923  FORMAT (1H+,79X,4F9.2)
9924  FORMAT (1H1//22(2H -)*LORAN TIME DIFFERENCE PREDICTIONS*21(2H- )/
     12X*IDENTIFIER*7X*LATITUDE*7X*LONGITUDE  ALTITUDE STEPSIZE --MEASUR
     2ED(USEC)- --COMPUTED(USEC)- ---ERROR(USEC)--- */
     312X,2(3X*(DEG-MIN-SEC)*),2X*(METER)*5X*(KM)*3(6X*TD1*6X*TD2*)/)
9925  FORMAT (//8(2H -)*MAP PARAMETERS*8(2H- )/
     12X*IDENTIFIER*7X*LATITUDE*7X*LONGITUDE*/
     212X,2(3X*(DEG-MIN-SEC)*)/)
9926  FORMAT (//11(2H -)*STATION PARAMETERS*11(2H- )/
     12X*IDENTIFIER*7X*LATITUDE*7X*LONGITUDE  EMISSION DELAY*/
     212X,2(3X*(DEG-MIN-SEC)*),3X*(USEC)*/)
      END



*DECK CORRAD
      SUBROUTINE CORRAD(RCOR,IDEG,IMIN,SEC,ID,IS,IERR)
      DIMENSION IDS(4)
      DATA (IDS=1HN,1HE,1HS,1HW)
      ISS = IS
      IF (ISS) 10,5,15
    5 IF (ID.EQ.IDS(1))  GO TO 25
      IF (ID.EQ.IDS(3))  GO TO 30
   10 IERR = 1
      RETURN
   15 IF (ISS-1) 20,20,10
   20 IF (ID.EQ.IDS(2))  GO TO 25
      IF (ID.EQ.IDS(4))  GO TO 30
      IERR = 1
      RETURN
   25 SIGN = 1.0
      GO TO 35
   30 SIGN = -1.0
   35 IF (IDEG-180) 40,40,10
   40 IF (IMIN-60) 45,10,10
   45 IF (SEC-60.0) 50,10,10
   50 RCOR = SIGN*(FLOAT(IDEG)*(1.7453292519943E-2)+FLOAT(IMIN)*
     + (2.9088820866572E-4)+SEC*4.8481368110953E-6)
      RETURN
      END
```

```
*DECK CORDMS
      SUBROUTINE CORDMS(RCOR,IDEG,IMIN,BEC,ID,IS,IERR)
      DIMENSION IDS(4)
      DATA (IDS=1HN,1HE,1HS,1HW)
      RANG = RCOR
      SEC = ABS(RANG)*206264.806247096
      ISS = IS
      IF (ISS) 5,10,15
    5 IERR = 1
      RETURN
   10 IF (SEC-324000.0005) 25,5,5
   15 IF (ISS-1) 20,20,5
   20 IF (SEC-648000.0005) 25,5,5
   25 IF (RANG) 30,35,35
   30 ISI = 2
      GO TO 40
   35 ISI = 0
   40 IDEG = SEC/3600.0
      IMIN = SEC/60.0-60.0*FLOAT(IDEG)
      SEC = SEC-3600.0*FLOAT(IDEG)-60.0*FLOAT(IMIN)
      ISEC = SEC
      SEC = SEC-FLOAT(ISEC)
      IF (SEC-0.9995) 60,45,45
   45 SEC = 0.0
      ISEC = ISEC+1
      IF (ISEC-60) 60,50,50
   50 ISEC = 0
      IMIN = IMIN+1
      IF (IMIN-60) 60,55,55
   55 IMIN = 0
      IDEG = IDEG+1
   60 LDX = ISI+ISS+1
      BEC=FLOAT(ISEC)+SEC
      ID=IDS(LDX)
      RETURN
      END
```

```
*DECK GEODI
      SUBROUTINE GEODI(B1, AL1, AZ12, S, B2, AL2, AZ21)
C PURPOSE:
C   CALCULATES INVERSE COMPUTATION FORM (SODANO, 1965).
C   SOUTHERN LATITUDES AND WESTERN LONGITUDES ARE NEGATIVE.
C INPUTS:
C   B1 = GEODETIC LATITUDE (RADIAN) OF FIRST POINT.
C   AL1 = GEODETIC LONGITUDE (RADIAN) OF FIRST POINT.
C   B2 = GEODETIC LATITUDE (RADIAN) OF SECOND POINT.
C   AL2 = GEODETIC LONGITUDE (RADIAN) OF SECOND POINT.
C OUTPUTS:
C   AZ12 = FORWARD AZIMUTH (RADIAN) FROM FIRST POINT.
C   AZ21 = BACK AZIMUTH (RADIAN) FROM SECOND POINT.
C   S = GEODETIC DISTANCE (METER) BETWEEN 2 POINTS.
      COMMON /PIS/TWOPI,PI,HAFPI,QRTPI
      COMMON /ZOTA/ARRAY(15)
      B0=ARRAY(5) $FL=ARRAY(6)
      IF (ABS(B1) .GT. QRTPI) GO TO 1
      TBET1 = TAN(B1) * (1. - FL)
      BET1 = ATAN(TBET1)
      GO TO 2
    1 CBET1 = 1.0 / TAN(B1) / (1. - FL)
      BET1 = ATAN(1. / CBET1)
    2 CONTINUE
      ALL1 = AL2 - AL1
      IF (AL2 - AL1 .EQ. 0.) 8, 9
    8 ALL2 = 0.
      GO TO 3
    9 CONTINUE
      ALL2 = AL2 - AL1 - SIGN (TWOPI, AL2 - AL1)
    3 IF (ABS(ALL1) .GT. ABS(ALL2)) 5, 6
    5 ALL = ALL2
      GO TO 7
    6 ALL = ALL1
    7 CONTINUE
   12 IF (ABS(ALL).EQ.0. .OR.ABS(ALL).EQ.PI .OR.ABS(ALL).EQ.TWOPI) 10,11
   10 ALL = ABS(ALL)
   11 CONTINUE
      IF (ABS(B2) .GT. QRTPI) GO TO 16
      TBET2 = TAN(B2) * (1. - FL)
      BET2 = ATAN(TBET2)
      GO TO 17
   16 CBET2 = 1.0 / TAN(B2) / (1. - FL)
      BET2 = ATAN(1. / CBET2)
   17 CONTINUE
      CBET1 = COS(BET1)
      SBET1 = SIN(BET1)
      CBET2 = COS(BET2)
      SBET2 = SIN(BET2)
      A = SBET1 * SBET2
      B = CBET1 * CBET2
      AB = SBET1 * CBET2
      BA = SBET2 * CBET1
      COSL = COS(ALL)
      SINL = SIN(ALL)
      CPHI = A + B * COSL
      SPHI = SQRT ((SINL * CBET2)**2 + (BA - AB * COSL)**2)
```

```
      IF (SPHI .GE. 0. .AND. CPHI .GE. 0.) 20, 21
20    PHI = ASIN(SPHI)
      IF (SPHI .GT. CPHI) PHI = ACOS(CPHI)
      GO TO 30
21    IF (SPHI .GE. 0. .AND. CPHI .LE. 0.) 22, 23
22    PHI = PI - ASIN(SPHI)
      IF (SPHI .GT. ABS(CPHI)) PHI = ACOS(CPHI)
      GO TO 30
23    IF (SPHI .LT. 0. .AND. CPHI .LT. 0.) 24, 25
24    PHI = PI - ASIN(SPHI)
      IF (ABS(SPHI) .GT. ABS(CPHI)) PHI = TWOPI - ACOS(CPHI)
      GO TO 30
25    IF (SPHI .LT. 0. .AND. CPHI .GE. 0.) 26, 27
26    PHI = TWOPI + ASIN(SPHI)
      IF (ABS(SPHI) .GT. CPHI) PHI = TWOPI - ACOS(CPHI)
      GO TO 30
27    CALL EXIT
30    CONTINUE
      C = B * SINL / SPHI
      FL2 = FL * FL
      CON1 = FL + FL2
      CON2 = 0.5 * FL2
      CON3 = SPHI * CPHI
      CON4 = PHI**2 / SPHI
      CON5 = CON4 * CPHI
      EM = 1. - C * * 2
      RATIO1 = (1. + CON1) * PHI + A * (CON1 * SPHI - CON2 * CON4)
     1 + EM * (-0.5 * CON1 * (PHI + CON3) + CON2 * CON5) - A * A * CON2
     2 * CON3 + EM * EM * CON2 * (0.125 * (PHI + CON3) - CON5 - 0.25 *
     3 CON3 * CPHI**2) + A * EM * CON2 * (CON4 + CON3 * CPHI)
      S = RATIO1 * B0
      IF (S .LE. 1.E-4) S = 0.
      RATIO2 = CON1 * PHI - A * CON2 * (SPHI + 2. * CON4)
     1 + 0.5 * EM * CON2 * (-5.0 * PHI + CON3 + 4.0 * CON5)
      ALAM = RATIO2 * C + ALL
      SALAM = SIN(ALAM)
      CALAM = COS(ALAM)
      CTAZ12 = BA - CALAM * AB
      CTAZ21 = BA * CALAM - AB
      IF (AL1 - AL2 .EQ. 0.)    35, 39
35    AZ12 = 0.
      AZ21 = 0.
      GO TO 34
39    CTAZ12 = CTAZ12 / (SALAM * CHET2)
      IF (CTAZ12 .EQ. 0.) 54, 55
54    AZ12 = HAFPI
      GO TO 56
55    CONTINUE
      AZ12 = ATAN(1. / CTAZ12)
56    CONTINUE
      CTAZ21 = CTAZ21 / (SALAM * CHET1)
      IF (CTAZ21 .EQ. 0.) 57, 58
57    AZ21 = HAFPI
      GO TO 34
58    CONTINUE
      AZ21 = ATAN(1. / CTAZ21)
34    CONTINUE
```

```
      IF (ALL .GE. 0. .AND. CTA712 .GE. 0.) 40, 41
   40 A712 = A712
      GO TO 50
   41 IF (ALL .GE. 0. .AND. CTA712 .LT. 0.) 42, 43
   42 A712 = PI + A712
      GO TO 50
   43 IF (ALL .LT. 0. .AND. CTA712 .GE. 0.) 44, 45
   44 A712 = PI + A712
      GO TO 50
   45 A712 = TWOPI + A712
   50 IF (ALL .GE. 0. .AND. CTA721 .GE. 0.) 46, 47
   46 A721 = PI + A721
      GO TO 51
   47 IF (ALL .GE. 0. .AND. CTA721 .LT. 0.) 48, 49
   48 A721 = TWOPI + A721
      GO TO 51
   49 IF (ALL .LT. 0. .AND. CTA721 .GE. 0.) 52, 53
   52 A721 = A721
      GO TO 51
   53 A721 = PI + A721
   51 CONTINUE
      A712 = AMOD(A712, TWOPI)
      IF (A712 .LT. 0.) A712 = A712 + TWOPI
      A721 = AMOD(A721, TWOPI)
      IF (A721 .LT. 0.) A721 = A721 + TWOPI
      RETURN
      END
```

```
*MODULE 2
*DECK GEORET
      SUBROUTINE GEORET(DSKM,NP)
      COMMON LAT(999),LON(999),MSINDX(5800)
      COMMON /GRND1/ZPP,ZD(6),Z(1009),ZPD(6),ZP(1009),XD(6),X(1009)
      COMMON /TE/ZPP1(1009),NPTS
      COMMON /SDRDI/S(999),DR(999),DI(999),LLM
      COMMON /CITCT/ITCT,LTOP,NPTCT,ZXMTR,ZMIN,ALTSW(3),TSW(3)
      NPTS=NP
      S(1)=0.
      X(1)=0.
      DO 1 I=1,NP
      CALL GETELV(LAT(I),LON(I),AAMP,FAZ,Z(I))
      IF(I.GT.1) S(I)=S(I-1)+DSKM
      DR(I) = AAMP * COS(FAZ)
      DI(I) = AAMP * SIN(FAZ)
    1 CONTINUE
      LLM=NP
C COMPUTE BASELINE TERRAIN ELEVATIONS RELATIVE TO TRANSMITTER ALTITUDE.
      ZXMTR=Z(1)
      Z(1)=0.
      DO 2 I=2,NP
C FOR NOW, CONVERT DISTANCE TO METERS.
      S(I)=S(I)*1000.
      X(I)=S(I)
      Z(I)=Z(I)-ZXMTR
    2 CONTINUE
C COMPUTE ELEVATION DERIVATIVES.
      NPM1=NP-1
      DO 3 L=2,NPM1
      CALL INT(L,L)
      ZPP1(L)=ZPP
    3 CONTINUE
      ZP(1)=0.
      ZPP1(1)=0.
      ZP(NP)=ZP(NP-1)
      ZPP1(NP)=ZPP1(NP-1)
      RETURN
      END
```

```
*DECK GEOPTS
      SUBROUTINE GEOPTS(SMB,RLATB,BLONR,RAZB)
      COMMON /PIS/TWOPI,PI,HAFPI,RTPI
      COMMON /PATH/RLATA,RLONA,RLATB,RLONB,RAZA,RAZB,SMB
      COMMON /ZDTA/ARRAY(15)
      AO = ARRAY(4) $ ESQ = ARRAY(7) $ CESQ = 1. - ESQ
      IF (ABS(RLATA)-HAFPI) 5,10,10
    5 IF (ABS(RLATB)-HAFPI) 15,10,10
   10 PRINT 9800,RLATA,RLATB
 9800 FORMAT(1H0, 89H     SUBROUTINE GEOPTS CALLED WITH END-POINT LATITUD
     +E OUT OF ACCEPTABLE RANGE (-PI/2,PI/2)//13X, 16HLATITUDE OF A  =,
     +F8.2, 29H RADIANS      LATITUDE OF B  =, F8.2, 8H RADIANS)
      PRINT 9801
 9801 FORMAT(33X, 28HPROGRAM EXECUTION TERMINATED)
       STOP
   15 IF (ABS(RLONA)-TWOPI) 20,20,25
   20 IF (ABS(RLONB)-TWOPI) 30,30,25
   25  PRINT 9802,RLONA,RLONB
 9802 FORMAT(1H0, 88H     SUBROUTINE GEOPTS CALLED WITH END-POINT LONGITU
     +DE OUT OF ACCEPTABLE RANGE (-2PI,2PI)//10X, 17HLONGITUDE OF A  =,
     +F9.2, 31H RADIANS      LONGITUDE OF B  =, F9.2, 8H RADIANS)
       PRINT 9801
       STOP
   30 ALONR=RLONA
       BLONR=RLONB
       IF(ALONR+PI) 35,35,40
   35 ALONR=ALONR+TWOPI
   40 IF(BLONR+PI) 45,45,50
   45 BLONR=BLONR+TWOPI
   50 BLON = BLONR-ALONR
      IF (BLON+PI) 55,70,75
   55 IF (BLONR) 60,65,65
   60 BLONR = BLONR+TWOPI
      GO TO 95
   65 ALONR = ALONR-TWOPI
      GO TO 95
   70 PRINT 9803
 9803 FORMAT(1H0, 88H     GEODESIC PATH INCLUDES A GEOGRAPHIC POLE - SUBR
     +OUTINE GEOPTS CANNOT HANDLE THIS CASE)
      PRINT 9801
      STOP
   75 IF (BLON-PI) 95,70,80
   80 IF (BLONR-PI) 90,90,85
   85 BLONR = BLONR-TWOPI
      GO TO 95
   90 ALONR = ALONR+TWOPI
   95 SMB = RLATA+RLATB
      BETA = SIN(0.5*SMB)
      BETA = (1.0-ESQ*BETA*BETA)/CESQ
      BLAT = AO/(BETA*SQRT(BETA*CESQ))
      BLON = BLONR-ALONR
      ALON = BETA*BLON
      Q = SIN(ALON)
      BLAT = COS(RLATB)
      ALAT = COS(RLATA)
      SAZA = Q*BLAT
      SAZB = Q*ALAT
```

```
      Q = SIN(0.5*ALON)
      Q = Q*Q
      AAZ = RLATB-RLATA
      CAZB = (1.0-Q)*SIN(AAZ)
      Q = Q*SIN(SMB)
      CAZA = CAZB+Q
      CAZB = CAZB-Q
      Q = SQRT(SAZA*SAZA+CAZA*CAZA)
      SMB = HLAT*ASIN(Q)
      SAZA = SAZA/Q
      CAZA = CAZA/Q
      SAZB = SAZB/Q
      CAZB = CAZB/Q
      RAZA = ATAN2(SAZA,CAZA)
      RAZB = ATAN2(SAZB,CAZB)
      ALON = RAZB-RAZA
      IF (ABS(ALON)-PI) 120,105,105
  105 IF (ALON) 110,110,115
  110 RAZB = RAZB+TWOPI
      GO TO 120
  115 RAZA = RAZA+TWOPI
  120 HLAT = CESQ+ESQ*ALAT*ALAT
      HLON = A0/SQRT(HLAT)
      HLAT = HLON*CESQ/HLAT
      HLON = HLON*ALAT
      CLAT = CAZA/HLAT
      CLON = SAZA/HLON
      HLAT = CESQ+ESQ*BLAT*BLAT
      HLON = A0/SQRT(HLAT)
      HLAT = HLON*CESQ/HLAT
      HLON = HLON*BLAT
      ALAT = CAZB/HLAT
      BLAT = (3.0*AAZ/SMB-ALAT-2.0*CLAT)/SMB
      ALON = SAZB/HLON
      BLON = (3.0*BLON/SMB-ALON-2.0*CLON)/SMB
      AAZ = 3.0*SMB
      ALAT = ((ALAT-CLAT)/SMB-2.0*BLAT)/AAZ
      ALON = ((ALON-CLON)/SMB-2.0*BLON)/AAZ
      AAZ = (RAZB-RAZA)/SMB
      ENTRY PCOORD
      RLATP = ((ALAT*SMP+BLAT)*SMP+CLAT)*SMP+RLATA
      RLONP = ((ALON*SMP+BLON)*SMP+CLON)*SMP+ALONR
      RAZP = AAZ*SMP+RAZA
      RETURN
      END
```

```
*DECK GETELV
      SUBROUTINE GETELV(LATIN,LONIN,AMP,FAZ,ELVTN)
C PURPOSE:
C  COMPUTES ALTITUDE AND IMPEDANCE AT A GIVEN LATITUDE AND LONGITUDE.
C  ASSUMES A DATABASE TAPE WITH 240 RECORDS PER ZONE OF POINTS, 60
C  WORDS PER RECORD, AND 2 DATA POINTS PER WORD. THESE GROUPINGS COVER
C  THE FOLLOWING MAP SECTIONS:
C   A ZONE SPANS 1 DEGREE LATITUDE (SOUTH TO NORTH) AND 2 DEGREES
C   LONGITUTDE (WEST TO EAST) OF THE MAP.
C   A RECORD SPANS 1 DEG. LAT. (S. TO N.) AND 30 SEC. LON. (W. TO E.)
C   OF A ZONE.
C   A WORD SPANS 1 MIN. LAT. (S. TO N.) AND 30 SEC. LON. (W. TO E.).
C   OF A RECORD.
C   A POINT LIES ON EVERY 30 SEC. LAT. AND EVERY 30 SEC. LON.
C  ALTITUDE IS INTERPOLATED WITHIN 4 POINTS WHICH LIE NORTHEAST,
C  SOUTHEAST, NORTHWEST, SOUTHWEST OF THE INPUT POINT.
C INPUTS:
C  LATIN = LATITUDE IN DDMMSS FORMAT.
C  LONIN = LONGITUDE IN DDDMMSS FORMAT.
C OUTPUTS:
C  AMP = IMPEDANCE AMPLITUDE.
C  FAZ = IMPEDANCE PHASE (RADIAN).
C  ELVTN = ALTITUDE (METER).
C CALLS ROUTINE INDEX FOR:
C  INDEX = POINTER TO A RECORD IN THE DATABASE.
C  IXPT = POINTER TO A HALF-WORD IN A RECORD IN THE DATABASE.
      COMMON /MAP/LATSW,LATNE,LONSW,LONNE
      COMMON /GETELV/L(120),IMP(120)
      COMMON /UNPACK/NPACK(60)
      COMMON /FLAGS/IXPT,NFLAG,NRPC,NRPDB
      COMMON LAT(999),LON(999),MSINDX(5800)
      DATA KPASS1,MASK8,MASK11 /1,377B,3777B/
      DATA INDDIM /5800/
C READ AND STORE DATABASE TAPE IN FIRST PASS ONLY.
      IF (KPASS1 .NE. 1) GO TO 630
      KPASS1=0
      NRPC=240*((LATNE-LATSW)/10000)
      NRPDB=NRPC*((LONNE-LONSW)/20000)
      IF (NRPDB+1 .LE. INDDIM) GO TO 610
      PRINT 9700, NRPDB $STOP
610    CALL OPENMS(2,MSINDX,INDDIM,0)
      DO 629 IRT=1,NRPDB
      BUFFERIN(13,1)(NPACK(1),NPACK(60))$IF(UNIT(13))620,629,620
620   CALL WRITMS(2,NPACK,60,IRT)
629   CONTINUE
C IF LATITUDE AND LONGITUDE LIE OUTSIDE MAP, PREVIOUS DATA ARE USED.
630   IF(LATIN.LT.LATSW  .OR.  LATIN.GE.LATNE) GO TO 190
      IF(LONIN.LT.LONSW  .OR.  LONIN.GE.LONNE) GO TO 190
C CONVERT TO RELATIVE LATITUDE AND LONGITUDE.
      NFLAG=0
      LATX=LATIN-LATSW
      IF (LATIN.GE.0) GO TO 200
C CORRECT FOR NEGATIVE LATITUDES.
      IF (MOD(LATX,10000).EQ.0) GO TO 200
      LATX=LATX-4000
      IF (MOD(LATX,100).EQ.0) GO TO 200
      LATX=LATX-40
```

```
C PROCEED WITH POSITIVE LATITUDES.
200    LATSEC=MOD(LATX,100)
       LATDM=LATX-LATSEC
       IF (LATSEC.GE.30) GO TO 240
       LATS=LATDM
       LATN=LATDM+30
       GO TO 260
240    LATS=LATDM+30
       LATN=LATDM+100
       LATM=MOD(LATDM,10000)
       IF (LATM.EQ.5900) LATN=LATDM+4100
260    LONX=LONIN-LONSW
       IF (LONIN.GE.0) GO TO 300
C CORRECT FOR NEGATIVE LONGITUDES.
       IF (MOD(LONX,10000).EQ.0) GO TO 300
       LONX=LONX-4000
       IF (MOD(LONX,100).EQ.0) GO TO 300
       LONX=LONX-40
C PROCEED WITH POSITIVE LONGITUDES.
300    LONSEC=MOD(LONX,100)
       LONDM=LONX-LONSEC
       IF (LONSEC.GE.30) GO TO 340
       LONW=LONDM
       LONE=LONDM+30
       GO TO 360
340    LONW=LONDM+30
       LONE=LONDM+100
       LONM=MOD(LONDM,10000)
       IF (LONM.EQ.5900) LONE=LONDM+4100
360    CONTINUE
C GET A DATA POINT NORTHEAST OF THE INPUT POINT.
       INDNE=INDEX(LATN,LONE)
       IF(NFLAG.EQ.1) GO TO 120
        CALL READMS(2,NPACK,60,INDNE)
       CALL UNPACK
       INE=L(IXPT)
C GET 3 POINTS SOUTHEAST, NORTHWEST, AND SOUTHWEST OF THE INPUT POINT.
       IF(MOD(LATN,10000).EQ.0.OR.MOD(LONE,20000).EQ.0)GOTO120
C ALL 4 POINTS LIE IN THE SAME ZONE.
       ISE=L(IXPT-1)
       CALL READMS(2,NPACK,60,INDNE-1)
       CALL UNPACK
       INW=L(IXPT)
       ISW=L(IXPT-1)
       GO TO 180
C THE 4 POINTS LIE IN DIFFERENT ZONES.
  120 CONTINUE
       INDNW=INDEX(LATN,LONW)
       IF(NFLAG.EQ.1) GO TO 140
       CALL READMS(2,NPACK,60,INDNW)
       CALL UNPACK
       INW=L(IXPT)
  140 CONTINUE
       INDSE=INDEX(LATS,LONE)
       IF(NFLAG.EQ.1) GO TO 160
       CALL READMS(2,NPACK,60,INDSE)
       CALL UNPACK
```

```
      ISE=L(IXPT)
  160 CONTINUE
      INDSW=INDEX(LATS,LONW)
      IF(NFLAG.EQ.1) GO TO 180
      CALL READMS(2,NPACK,60,INDSW)
      CALL UNPACK
      ISW=L(IXPT)
  180 CONTINUE
      AMP=FLOAT(SHIFT(IMP(IXPT),-11).A.MASK8)*.001
      FAZ=FLOAT(IMP(IXPT).A.MASK11)*.001
      IF(NFLAG.EQ.1) PRINT 9702
      ELEAST=FLOAT(ISE)+FLOAT((INE-ISE)*MOD(LATSEC,30))/30.
      ELWEST=FLOAT(ISW)+FLOAT((INW-ISW)*MOD(LATSEC,30))/30.
      ELVTN=ELWEST+(ELEAST-ELWEST)*FLOAT(MOD(LONSEC,30))/30.
      RETURN
  190 PRINT 9701, LATIN,LONIN
      RETURN
 9700 FORMAT (* DATABASE HAS *I10* RECORDS. INCREASE ARRAY MSINDX AND VA
     1RIABLE INDDIM TO ACCOMMODATE. RUN ABORTS.*)
 9701 FORMAT(* *,4H****,*LAT=*I10*,LON=*I10* REQUESTED ARE OUTSIDE MAP R
     1EGION - NO ELEVATION RETURNED*,//)
 9702 FORMAT(* *,4H****,*INDEX REQUESTED EXCEEDS SCANLINES GENERATED. TH
     1US PREVIOUS ELEVATIONS WILL BE USED*)
      END
```

```
*DECK INDEX
      FUNCTION INDEX(LAT,LON)
C PURPOSE:
C  COMPUTES POINTERS INTO DATABASE FOR A GIVEN LATITUDE AND LONGITUDE.
C INPUTS:
C  LAT = RELATIVE LATITUDE IN DDMMSS FORMAT.
C  LON = RELATIVE LONGITUDE IN DDDMMSS FORMAT.
C  NRPC = NUMBER OF RECORDS PER COLUMN OF MAP AREA 2 DEGREES LONGITUDE
C         WIDE.
C  NRPDB = NUMBER OF RECORDS IN THE DATABASE.
C OUTPUTS:
C  INDEX = POINTER TO RECORD IN DATABASE.
C  IXPT = POINTER TO DATA POINT (= HALF-WORD) IN DATABASE RECORD.
C  NFLAG = 1 INDICATES COMPUTED RECORD POINTER IS OUT OF RANGE.
      COMMON /FLAGS/IXPT,NFLAG,NRPC,NRPDB
      LATS=MOD(LAT,100)
      LATM=MOD(LAT,10000)-LATS
      LATD=(LAT-LATM-LATS)/10000
      LATM=LATM/100
      LONS=MOD(LON,100)
      LONM=MOD(LON,10000)-LONS
      LOND=(LON-LONM-LONS)/10000
      LONM=LONM/100
      ILOND=(LOND/2)*NRPC+MOD(LOND,2)*120+LONM*2+LONS/30+1
      INDEX=ILOND+LATD*240
      IXPT=LATM*2+LATS/30+1
      IF(INDEX.GT.NRPDB)NFLAG=1
      RETURN
      END
```

```
*DECK UNPACK
      SUBROUTINE UNPACK
      COMMON /GETELV/L(120),IMP(120)
      COMMON /UNPACK/NPACK(60)
      DATA MASK11/3777B/,MASK19/1777777B/
      DO 20 MA=1,60 $M=2*(MA-1)+1 $NPKMA=NPACK(MA)
      L(M)=AND(MASK11,SHIFT(NPKMA,-49))*8
      IMP(M)=AND(MASK19,SHIFT(NPKMA,30))
      L(M+1)=AND(MASK11,SHIFT(NPKMA,-19))*8
      IMP(M+1)=AND(MASK19,NPKMA)
   20 CONTINUE
      RETURN
      END
```

```
*DECK INT
      SUBROUTINE INT (I,K)
C INPUTS:
C   I = POSITION IN X AND Z ARRAYS ON WHICH TO CENTER CALCULATIONS
C   K = POSITION IN ARRAYS Z AND ZP TO STORE CALCULATED VALUES
      COMMON /GRND1/ZPP,ZD(6),Z(1009),ZPD(6),ZP(1009),XD(6),X(1009)
      IMO=I-1 $XIMO=X(IMO) $ZIMO=Z(IMO) $ZI=Z(I)
      IPO=I+1 $XIPO=X(IPO) $XI=X(I) $XK=X(K)
      C=((Z(IPO)-ZIMO)-(ZI-ZIMO)*(XIPO-XIMO)/(XI-XIMO))/
     1 ((XIPO*XIPO-XIMO*XIMO)-(XI*XI-XIMO*XIMO)*
     2 (XIPO-XIMO)/(XI-XIMO))
      B=((ZI-ZIMO)-C*(XI*XI-XIMO*XIMO))/(XI-XIMO)
      A=ZI-XI*(B+C*XI)
      Z(K)=A+XK*(B+C*XK)
      ZP(K)=B+2.*C*XK
      ZPP=2.*C
      RETURN
      END
```

```
*MODULE 3
*DECK INEG2F
      SUBROUTINE INEG2F
C REFERENCES:
C    (1) JOHLER,J.R., BERRY,L.A., LORAN-D PHASE CORRECTIONS OVER
C          INHOMOGENEOUS, IRREGULAR TERRAIN., ESSA TECHNICAL REPORT
C          IER 59-ITSA 56, NOVEMBER, 1967.
      COMMON /PIS/TWOPI,PI,HAFPI,QRTPI
      COMMON /GRND1/ZPP,ZD(6),Z(1009),ZPD(6),ZP(1009),XD(6),X(1009)
      COMMON /GRND2/RO,DEN,R,U,OMX,R2,DELI
      COMMON /TE//PP1(1009),NPTS
      COMMON /ZOTA/ARRAY(15)
      COMMON /SORDI/S(999),OR(999),DI(999),LLM
      COMMON /CITCT/ITCT,LTOP,NPTCT,ZXMTR,ZMIN,ALTSW(3),TSW(3)
C ZD, ZPD AND XD PROVIDE STORAGE FOR NEGATIVE INDEXES.
      DIMENSION TZER(1),T(1000),GW(5),GX(5),EW(9),EX(9),WG(5),XS(9)
      DIMENSION WE(9),W(1000)
      COMPLEX DELI,DEN,FLEAF,F1,F2,F3,G1,G2,G3,SUM,TERM,
     1 W,WE,WG,WW,WX,TS,FIND,FI
      DATA KPASS1/1/
      DATA ((GX(K), K = 1, 5) = .02216356881, .1678315676, .4615973615,
     1.7483346284, .9484939262)
      DATA ((GW(K), K = 1, 5) = .5910484494, .5385334386, .4381772725,
     1 .2989026983, .1333426886)
      DATA ((EX(K), K = 1, 4) = .1051402826, .3762246145, .6694480124,
     1 .9373342493)
      DATA ((EW(K), K = 1, 4) = .06568051989, .1960962655, .2525273456,
     1.1523625357)
      DATA (TZER(1)=1.0)
9400  FORMAT (* AIRBORNE TARGET ALTITUDE (*F9.2*) LIES BELOW TERRAIN ELE
     1VATION (*F9.2*). RUN ABORTS.*)
      FI(WAVER) = CMPLX(COS(WAVER), SIN(WAVER))
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C START FIRST-PASS INITIALIZATION.
      IF (KPASS1 .NE. 1) GO TO 9999
      KPASS1=0
C FREQ IS USED IN KHZ IN THIS PROGRAM.
      FREQ = ARRAY(1)*1.E-3 $RAD = ARRAY(13) $FLAT = ARRAY(11)
      FORH = ARRAY(14) $SMOOTH = ARRAY(15) $ETA = ARRAY(2)
      NZERO=0
      T(NZERO)=1.0
C WAVE NUMBER IN ATMOSPHERE AT SURFACE OF EARTH
      WAVE = ARRAY(10)
      AMICRO = 1.0/(TWOPI*FREQ)
      TX = SQRT(FREQ*ETA)*.0408389549
C INITIALIZE QUANTITIES
      X(1) = 0.
      W(1) = 1.
C TREAT SMOOTH ROUND EARTH IF SMOOTH=1.
      IF (SMOOTH .EQ. 1.) Z(1)=ZP(1)=ZPP1(1)=0.
 9999 CONTINUE
C END FIRST-PASS INITIALIZATION.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C START INITIALIZATION FOR CURRENT PATH.
C COMPUTE DISTANCE AND DISTANCE INCREMENT (KM) TO TARGET.
      DMAX=X(NPTS)/1000.
      DINC=X(2)/1000.
```

```
      LTOP = 0
      NOFLAT = 1
C IMPEDANCE SHOULD BE REFERENCED TO 100 KHZ
C FREQUENCY DEPENDENCE OF IMPEDANCE  = *SQRT(FREQ/0.1)
      DELI = CMPLX(DR(1), DI(1))*SQRT(FREQ/0.1)
      IMOST = DMAX/DINC + .01
      DELTX = 1000.*DINC
      T(1) = 1./SQRT(DELTX)
C FLAT EARTH THEORY DESCRIBED IN REFERENCE 3
      DO 32 K = 1, 5
C ATTENUATION FOR FLAT EARTH THEORY
      X(-K) = DELTX*GX(K)
      CALL GROUND ( -K, 2, 0, 0)
   32 WG(K) = FLEAF(WAVE, 0., 0., X(-K), DELI)
      I = 2
C END INITIALIZATION FOR CURRENT PATH.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C START LOOP ON SURFACE POINTS ALONG HUFFORD BASELINE.
   37 CONTINUE
C TREAT SMOOTH ROUND EARTH IF SMOOTH=1.
      IF (SMOOTH .EQ. 1.) Z(I)=ZP(I)=ZPP1(I)=0.
      ZPP=ZPP1(I)
      T(I) = 1./SQRT(X(I) + DELTX)
      OMX = OMCOS(X(I)/RAD)
      IGO = 2 - MOD(I, 2)
      IL = I - IGO - 1
C EQUATION 2.18, REFERENCE 1
      RO = SQRT(2.*RAD*(RAD + Z(I))*OMX + Z(I)**2)
      IF (NOFLAT .EQ. 2) GO TO 45
C FLAT DETERMINES DISTANCE TO WHICH FLAT EARTH THEORY IS USED BEFORE
C IRREGULAR, INHOMOGENEOUS TERRAIN MAY BE INSERTED.
   40 IF(I.LE.4.OR.FLAT.GT.X(I)) 41,45
   41 W(I) = FLEAF(WAVE, 0., 0., X(I), DELI)
      GO TO 90
C SPHERICAL EARTH THEORY
   45 SUM = 0.
      NOFLAT = 2
      DO 50 K = 1, 5
C R, DEN AND U ARE FROM GROUND
      CALL GEOM(I, -K, 1, 0.)
C EQUATION 2.27, REFERENCE 1
      TERM = WG(K)*EI(-WAVE*R)*DEN
   50 SUM = (U*GW(K))/SQRT(X(I) - X(-K))*TERM + SUM
C SIMPSONS RULE
      SUM = 3.*T(1)*SUM
      KK = 1
      IF (IL .LT. 3) GO TO 100
      DO 60 K = 3, IL
      CALL GEOM(I, K, 1, 0.)
      TERM = U*T(K-1)*W(K)*EI(-WAVE*R)*DEN
      IF (KK .EQ. 2) GO TO 55
      SUM = 4.*T(I-K)*TERM + SUM
      KK = 2
      GO TO 60
   55 SUM = 2.*T(I-K)*TERM + SUM
      KK = 1
   60 CONTINUE
```

```
 100   CONTINUE
       CALL GEOM(I, 2, 1, 0.)
       SUM = T(I-2)*T(1)*U*W(2)*EI(-WAVE*R)*DEN + SUM
       CALL GEOM(I, I - IGO, 1, 0.)
       F2 = U*T(IL)*W(I-IGO)*EI(-WAVE*R)*DEN
       SUM = (SUM + T(IGO)*F2)*.333333333*DELTX
       IF (IGO .EQ. 2) GO TO 66
       F1 = F2
       F2 = TERM
       GO TO 70
  66   CALL GEOM(I, I - 1, 1, 0.)
       F1 = U*T(I-2)*W(I-1)*EI(-WAVE*R)*DEN
       SUM = SUM + .083333333*DELTX*(5.*T(1)*F1 + 8.*T(2)*F2
      1        - T(3)*TERM)
  70   Q = TX/T(I-1)
       RHO = 1. + ZP(I)**2
       RHO = ZPP/RHO
       TERM = 1.2/T(1)*T(I-1)*CMPLX(Q,Q)*(DELI + CMPLX(0., -RHO/WAVE))
       WX = 1. - CMPLX(Q,Q)*(SUM + 2./T(1)*(.466666667*F1
      1        - .066666667*F2))/(1. + TERM)
       W(I) = WX
  90   DIST = .001*X(I)
       CALL INDF (0.,X(I),EORH,FIND)
       TS=W(I)*FIND
       PHA=CANG(TS)
C PHIC IS THE SECONDARY PHASE CORRECTION IN RADIANS.
       PHIC = - (PHA - WAVE*(R0 - X(I)))
C SEC IS THE SECONDARY PHASE CORRECTION IN MICROSECONDS.
       SEC = PHIC*AMICRO
C SAVE SECONDARY PHASE FACTOR AT TARGET GROUND POINT.
       IF(I.NE.NPTCT) GO TO 987
       TSW(ITCT)=SEC
987    CONTINUE
       I = I + 1
       IF(I.LE.NPTS)  GO TO 37
C END LOOP ON SURFACE POINTS ALONG HUFFORD BASELINE.
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
C START AIRBORNE TARGET COMPUTATION.
       IF (ZMIN.LE.0.)    RETURN
       LTOP = 1
       F2=(0.,0.)
       G2=(0.,0.)
       OLDR=0.
       II = I = NPTCT
       IGO = 2 - MOD(I, 2)
       DO 155 K = 1, 4
       X(1000 + K) = X(I) - DELTX*FX(K)
       XS(K) = X(1000 + K)
 155   CALL GROUND (1000+K,II,0,0)
       CALL CHEVKEN(X, W, IMOST, XS, HE, 4, 5, 1)
       HL = ZMIN - ZXMTR
C ABORT IF TARGET ALTITUDE IS BELOW TERRAIN.
       IF (HL .GT. Z(I)) GO TO 154
       PRINT 9400,HL,Z(I) $STOP
 154   ZZ = RAD + HL
       R0 = SQRT(2.*RAD*ZZ*ONCOS(X(I)/RAD) + HL**2)
       SUM = 0.
```

```
      DO 157 K = 1, 5
      CALL GEOM(I, -K, 2, HL)
      TERM = XG(K)*EI(-WAVE*R)*DEN
  157 SUM = U*GW(K)*TERM/SQRT(X(I) - X(-K)) + SUM
C SIMPSONS RULE
      SUM = 3.*T(1)*SUM
      CALL GEOM(I, 3, 2, HL)
      F1 = T(1)*T(I-2)*U*DEN*W(2)
      G1 = EI(-WAVE*R)
      SUM = SUM + F1*G1
      ILO = I - IGO
      KGO = 1
      IF (ILO .LT. 3) GO TO 115
      DO 182 K = 3, ILO
      F3 = F2
      G3 = G2
      F2 = F1
      G2 = G1
      CALL GEOM(I, K, 2, HL)
      G1 = EI(-WAVE*R)
      F1 = T(K-1)*T(I-K)*U*DEN*W(K)
      DELTG = WAVE*(R - OLDR)
      IF (KGO-2) 158, 159, 172
  158 SUM = SUM + 4.*F1*G1
      KGO = 2
      GO TO 182
  159 SUM = SUM + 2.*F1*G1
      KGO = 1
      IF (ABS(DELTG) .GT. .2) 170, 182
  170 KGO = 3
      SUM = SUM - F1*G1
      GO TO 182
C EQUATION 2.35, REFERENCE 1
  172 SUM = SUM + CMPLX(0., 3./DELTG)*(G1*(F1 - .5*(F1*CMPLX(2
     1., 3.*DELTG) - 4.*F2*CMPLX(1., DELTG) + F3*CMPLX(2., DELTG
     2))/DELTG**2) - G2*(F2 - .5*(F1*CMPLX(2., DELTG) - 4.*
     3F2 + F3*CMPLX(2., -DELTG))/DELTG**2))
  182 OLDR = R
  115 CONTINUE
      IF (IGO .EQ. 1) GO TO 188
      F3 = F2
      G3 = G2
      F2 = F1
      G2 = G1
      CALL GEOM(I, I - 1, 2, HL)
      G1 = EI(-WAVE*R)
      F1 = T(I-2)*T(1)*U*DEN*W(I-1)
      DELTG = WAVE*(R - OLDR)
      IF (ABS(DELTG) .GT. .2) 184, 185
C EQUATION 2.35, REFERENCE 1
  184 SUM = SUM + CMPLX(0., 3./DELTG)*(G1*(F1 - .5*(F1*CMPLX(2
     1., 3.*DELTG) - 4.*F2*CMPLX(1., DELTG) + F3*CMPLX(2., DELTG
     2))/DELTG**2) - G2*(F2 - .5*(F1*CMPLX(2., DELTG) - 4.*
     3F2 + F3*CMPLX(2., -DELTG))/DELTG**2))
      GO TO 188
  185 SUM = SUM + 1.25*F1*G1 + 2.*F2*G2 - .25*F3*G3
  188 DO 190 K = 1, 4
```

```
      CALL GEOM(I, 1000 + K, 3, HL)
      SUM = SUM + 3.*FW(K)*VE(K)*EI(-WAVE*R)*DEN*U/(SQRT(XS(K))*T(1))
  100 CONTINUE
      SUM = .3333333*DELTX*SUM
      Q = TX/T(I - 1)
      WV = (1. - CMPLX(0.,Q)*SUM)*.5
      CALL INDF (HL,X(I),EORM,FIND)
      TS=WV*FIND
      FRAD = - (CANG(TS) - WAVE*(R0 - X(I)))
C FMSEC IS THE PHASE IN MICROSECONDS WHILE FRAD IS THE PHASE IN RADIANS
      FMSEC = FRAD*AMICRO
C SAVE PHASE AT TARGET POINT FOR TIME DIFFERENCE CALCULATION.
      ALTSV(ITCT) = FMSEC
      RETURN
      END
```

```
*DECK GROUND
      SUBROUTINE GROUND(I, K, IGO, HH)
C THIS SUBROUTINE USES VARIABLE IMPEDANCE AND TERRAIN TO CALCULATE
C GROUND WAVE FIELDS
      COMMON /GRND1/ZPP,ZD(6),Z(1009),ZPD(6),ZP(1009),XD(6),X(1009)
      COMMON /GRND2/R0,DEN,R,U,OMX,R2,DELTA
      COMMON /ZOTA/ARRAY(15)
      COMMON /SDRDI/S(999),DR(999),DI(999),LLM
      COMPLEX DELTA,DEN
C
      CALL INT (K,I)
      FREQ = ARRAY(1)*1.E-3 $WAVE = ARRAY(10) $A = ARRAY(13)
      RETURN
C
      ENTRY GEOM
      HIT = Z(I) + HH
      IF (I .EQ. K) GO TO 20
      T = (X(I) - X(K))/A
      GS = A + Z(K)
      GX = A + HIT
      CT = COS(T)
      ST = SIN(T)
      OT = OMCOS(T)
C EQUATION 2.23, REFFERENCE 1
      R2 = SQRT(2.*GS*GX*OT + (HIT - Z(K))**2)
C EQUATION 2.22, REFFERENCE 1
C EQUATION 2.30, REFFERENCE 1
      R1 = SQRT(2.*A*GS*OMCOS(X(K)/A) + Z(K)**2)
      R = R1 + R2 - R0
C EQUATION 2.30, REFERENCE 1
C U = PROJECTION FACTOR TO THE SURFACE OF INTEGRATION.
      U = X(K)*R0*SQRT(1. + ZP(K)**2)/(R1*R2*X(I))
      IF (IGO .LT. 3) U = U*(X(I) - X(K))
C EQUATION 2.30, REFERENCE 1
C PD = NORMAL PARTIAL DERIVATIVE TO THE SURFACE OF THE GROUND.
      PD = (A*OT + Z(K) - HIT*CT + GX*ZP(K)*A*ST/GS)/R2
      XK = X(K)
      IF (XK .GE. S(LLM)) GO TO 12
      LLL = 1
      IF (XK .LT. S(1)) GO TO 10
      LLMO = LLM - 1
      IF (LLMO .LT. 1) GO TO 100
      DO 13 LLL = 1, LLMO
      IF (XK .LT. S(LLL + 1) .AND. XK .GE. S(LLL)) GO TO 10
   13 CONTINUE
  100 CONTINUE
      GO TO 21
C FREQUENCY DEPENDENCE OF IMPEDANCE   = *SQRT(FREQ/0.1)
C DELTA = COMPLEX LOCAL GROUND IMPEDANCE THAT IS ADDED TO A
C FUNCTION INVOLVING WAVE, R2 AND PD.
   10 DELTA = CMPLX(DR(LLL), DI(LLL))*SQRT(FREQ/0.1)
      GO TO 21
C FOR VALUES OF X GREATER THAN THE LAST S VALUE READ IN, THE LAST
C IMPEDANCE IS USED
C FREQUENCY DEPENDENCE OF IMPEDANCE   = *SQRT(FREQ/0.1)
   12 DELTA = CMPLX(DR(LLM), DI(LLM))*SQRT(FREQ/0.1)
   21 CONTINUE
```

```
C DELTA2 = DERIVATIVE PART OF IMPEDANCE
C EQUATION 21. REFERENCE 2
      DELTA2 = CMPLX(1., -1./(WAVE*R2))*PD
C EQUATION 20. REFERENCE 2
   19 DEN = DELTA2 + DELTA
      RETURN
   20 U = 0.
      R2 = GX - A - Z(I)
      RETURN
      END
```

```
*DECK OMCOS
      FUNCTION OMCOS(X)
C OMOCOS(X)= 1 - COS(X)
      IF (ABS(X) .GT. .15) GO TO 40
      IF (X .EQ. 0.) GO TO 50
      S = X*X
      OMCOS = T = .5*S
      R = 4.
   10 T = -T*S/(R*(R - 1.))
      OMCOS = OMCOS + T
      IF (T/OMCOS .LE. .5E-9) GO TO 51
      R = R + 2.
      GO TO 10
   40 OMCOS = 1. - COS(X)
      RETURN
   50 OMCOS = 0.
   51 RETURN
      END
```

```
*DECK CNEVKEN
      SUBROUTINE CNEVKEN(A, FA, NA, X, FX, NX, NPT, KASE)
C INTERPOLATION OF F(X) FOR AN X ARRAY USING AITKEN*S METHOD FOR
C EXTRAPOLATION OR INTERPOLATION NEAR THE ENDS OF THE GIVEN ARRAY AN
C NEVILLE*S METHOD FOR ALL OTHER INTERPOLATION.
C INPUT
C    A = THE ARRAY OF GIVEN ABSCISSA IN ASCENDING ORDER.
C    FA = THE ARRAY OF GIVEN VALUES OF F(A). FA IS COMPLEX.
C    NA = THE NUMBER OF ELEMENTS IN EACH OF THE ARRAYS A AND F(A).
C    X = THE VALUES OF X IN ASCENDING ORDER FOR WHICH ONE WISHES TO
C        DETERMINE F(X).
C    NX = THE NUMBER OF ELEMENTS IN EACH OF THE ARRAYS X AND F(X).
C    NPT = THE ODD NUMBER OF POINTS USED FOR INTERPOLATION.  NPT CAN
C        NOT BE GREATER THAN NA.
C    KASE = 1, THE PROGRAM PROCEEDS NORMALLY
C    KASE = 2, F(X) IS CALCULATED ONLY FOR THOSE X VALUES THAT REQUIR
C    A(NA) FOR INTERPOLATION
C OUTPUT
C    FX = THE ARRAY OF VALUES APPROXIMATED FOR F(X). FX IS COMPLEX.
      COMPLEX FA, FX, FUNCT, POLY
      DIMENSION A(NA), FA(NA), X(NX), FX(NX), FUNCT(15), ABSC(15), DIF(1
     15), POLY(15)
9600  FORMAT (1H0/50H THE X VALUES ARE NOT ARRANGED IN ASCENDING ORDER./
     1/5X,4HI = E20.9,5X,7HX(I) = E20.9,5X,4HJ = E20.9,5X,7HA(J) = E20.9
     2///14X,1HX,20X,4HF(X))
9601  FORMAT (5X,2E20.9)
9602  FORMAT (1H0* THERE ARE NOT ENOUGH POINTS IN THE GIVEN ARRAY.*)
      LOOP = 1
      IF (NPT - 15) 3, 3, 9
    9 NPT = 15
    3 IF (NPT - NA) 8, 8, 4
    4 NPT = NPT - 2
      IF (NPT - 1) 5, 5, 6
    5 PRINT 9602
      CALL EXIT
    6 IF (LOOP .EQ. 2) GO TO 3
    7 LOOP = 2
      GO TO 3
    8 NPT2 = NPT/2
      IF (KASE .EQ. 2) GO TO 12
   11 NSTART = 1
      MX = 1
      GO TO 16
   12 NSTART = NA - 3
      TEST = A(NSTART) + (A(NSTART + 1) - A(NSTART))/2
      IF (NX .LT. 1) GO TO 170
      DO 14 I = 1, NX
      IF (X(I) - TEST) 14, 14, 13
   13 MX = I
      GO TO 16
   14 CONTINUE
  170 CONTINUE
   16 NSTOP = NA - 1
      IF (NX .LT. MX) GO TO 175
      DO 125 I = MX, NX
      IF (X(I) - A(1)) 135, 15, 10
   10 IF (X(I) - A(NA)) 25, 20, 130
```

```
 15  FX(I) = FA(1)
     GO TO 125
 20  FX(I) = FA(NA)
     GO TO 125
 25  IF (NSTOP .LT. NSTART) GO TO 180
     DO 85 J = NSTART, NSTOP
     IF (X(I) - A(J)) 32, 35, 30
 30  IF (X(I) - A(J + 1)) 45, 40, 85
 32  II = I - 1
     PRINT 9600, I, X(I), J, A(J)
     IF (II) 34, 34, 33
 33  PRINT 9601, (X(N), FX(N), N = 1, II)
 34  CALL EXIT
 35  FX(I) = FA(J)
     NSTART = J
     GO TO 125
 40  FX(I) = FA(J + 1)
     NSTART = J + 1
     GO TO 125
 45  NSTART = J
     IF (ABS(X(I) - A(J)) - ABS(X(I) - A(J + 1))) 50, 50, 55
 50  JJ = J
     GO TO 60
 55  JJ = J + 1
     GO TO 60
 85  CONTINUE
180  CONTINUE
 60  IF (JJ - NPT2) 135, 135, 70
 70  IF (JJ + NPT2 - NA) 80, 80, 130
 80  KK = JJ - NPT2 - 1
 90  IF (NPT .LT. 1) GO TO 185
     DO 95 K = 1, NPT
     KK = KK + 1
     FUNCT(K) = FA(KK)
     ABSC(K) = A(KK)
 95  DIF(K) = ABSC(K) - X(I)
185  CONTINUE
     NTOP = NPT - 1
     LL = 1
100  IF (NTOP .LT. 1) GO TO 190
     DO 105 L = 1, NTOP
     LLL = L + LL
105  POLY(L) = (FUNCT(L)*DIF(LLL) - FUNCT(L+1)*DIF(L))/(ABSC(LLL)
    1 - ABSC(L))
190  CONTINUE
     IF (NTOP - 1) 120, 120, 110
110  DO 115 M = 1, NTOP
115  FUNCT(M) = POLY(M)
     NTOP = NTOP - 1
     LL = LL + 1
     GO TO 100
130  INC = - 1
     KK = NA + 1
     GO TO 140
135  INC = 1
     KK = 0
140  IF (NPT .LT. 1) GO TO 215
```

```
      DO 145 K = 1, NPT
      KK = KK + INC
      FUNCT(K) = FA(KK)
      ABSC(K) = A(KK)
  145 DIF(K) = ABSC(K) - X(I)
  215 CONTINUE
      NTOP = NPT - 1
      LL = 1
  150 IF (NTOP .LT. 1) GO TO 220
      DO 155 L = 1, NTOP
      LLL = L + LL
  155 POLY(L) = (FUNCT(L)*DIF(LLL) - FUNCT(L+1)*DIF(LL))/(ABSC(LLL)
     1 - ABSC(LL))
  220 CONTINUE
      IF (NTOP - 1) 120, 120, 160
  160 DO 165 M = 1, NTOP
  165 FUNCT(M) = POLY(M)
      NTOP = NTOP - 1
      LL = LL + 1
      GO TO 150
  120 FX(I) = POLY(1)
  125 CONTINUE
  175 CONTINUE
      RETURN
      END
```

```
*DECK FLEAF
      COMPLEX FUNCTION FLEAF(WAVE, H1, H2, XD, DELTAR)
C FLAT EARTH THEORY, REFFRENCE 3
      COMPLEX TEMP, Q, Z, Z2, ZZ, HWERF, WERFZ, WERF, ZWERF, DELTAR
      HD = H2 - H1
      TEMP = (0.7071067812, - 0.7071067812)*SQRT(.5*WAVE)
      XD2 = SQRT(XD)
      Q = -TEMP*HD/XD2
      Z = TEMP*DELTAR*XD2 + Q
      ZZ = - Z
      ZI = AIMAG(ZZ)
      IF (ZI.LT.0. .OR. (ABS(REAL(ZZ)).LT.6. .AND. ZI.LT.6.)) GO TO 10
      Z2 = ZZ**2
      HWERF = (Z2 - 2.)/(ZZ*(Z2 - 3.5))
      GO TO 12
C WERF = COMPLEMENTARY ERROR FUNCTION
   10 WERFZ = WERF(ZZ)
      HWERF = ZZ - 0.5*WERFZ/(ZZ*WERFZ + (0., - 0.56418958))
   12 ZWERF = Z + HWERF
      FLEAF = (Q*ZWERF - 0.5)/(Z*ZWERF - 0.5)
      RETURN
      END
```

```
*DECK WERF
      COMPLEX FUNCTION WERF(ZZZ)
C COMPLEMENTARY ERROR FUNCTION
      COMPLEX Z, ZZZ, ZV, V, Z2, C, W, S
      DIMENSION C(12), W(5, 4)
      EQUIVALENCE (S, C(12)),
      DATA (C(1) = (.0, - .5641895835))
      DATA W/(1.,0.0),
     1 (3.678794411714423E-01,6.071577058413937E-01),
     2 (1.831563888873418E-02,3.400262170660662E-01),
     3 (1.234098040866788E-04,2.011573170376004E-01),
     4 (1.125351747192646E-07,1.459535899001528E-01),
     5 (4.275835761558070E-01,0.000000000000000E+00),
     6 (3.047442052569126E-01,2.082189382028316E-01),
     7 (1.402395813662779E-01,2.222134401798991E-01),
     8 (6.531777728904697E-02,1.739183154163490E-01),
     9 (3.628145648998864E-02,1.358389510006551E-01),
     A (2.553956763105058E-01,0.000000000000000E+00),
     B (2.184926152748907E-01,9.299780939260186E-02),
     C (1.479527595120158E-01,1.311797170842178E-01),
     D (9.271076642644332E-02,1.283169622282615E-01),
     E (5.968692961044590E-02,1.132100561244882E-01),
     F (1.790011511813930E-01,0.000000000000000E+00),
     G (1.642611363929861E-01,5.019713513524966E-02),
     H (1.307574696698522E-01,8.111265047745472E-02),
     I (9.640250558304439E-02,9.123632600421258E-02),
     J (6.979096164964750E-02,8.934000024036461E-02)/
      XX = REAL(ZZZ)
      YY = AIMAG(ZZZ)
      X = ABS(XX)
      Y = ABS(YY)
      Z = CMPLX(X, Y)
      LZ2 = 0
      IF (X .GE. 4.5 .OR. Y .GE. 3.5) GO TO 100
      I = X + .5
      J = Y + .5
      V = CMPLX(FLOAT(I), FLOAT(J))
      ZV = Z - V
      C(2) = W(I + 1, J + 1)
      AI = 0.
      DO 10 I = 3, 12
      AI = AI - .5
      C(I) = (V*C(I - 1) + C(I - 2))/AI
   10 CONTINUE
      J = 12
      DO 11 I = 2, 11
      J = J - 1
   11 S = S*ZV + C(J)
   20 IF (YY .GE. 0.) GO TO 30
      IF (LZ2 .EQ. 0) Z2 = Z*Z
      S = 2.*CEXP(-Z2) - S
      IF (XX .GT. 0.) S = CONJG(S)
      GO TO 200
   30 IF (XX .LT. 0.) S = CONJG(S)
  200 WERF = S
      RETURN
  100 LZ2 = 1
```

```
      Z2 = Z*Z
      S = Z*((0.,0.4613135279)/(Z2 - 0.1901635092) + (0.,0.09999216168)
     1/(Z2 - 1.7844927485) + (0.,0.0028838938748)/(Z2 - 5.5253437437 9))
      GO TO 20
      END
```

```
*DECK CANG
      FUNCTION CANG(Z)
C PURPOSE:
C  COMPUTES THE ARGUMENT OF A COMPLEX NUMBER Z SUCH THAT
C  -PI .LT. THETA .LE. PI.
      COMMON /PIS/TWOPI,PI,HAFPI,SRTPI
      COMPLEX Z
      X=REAL(Z)
      Y=AIMAG(Z)
      IF (X) 20,30,10
   10 CANG=ATAN2(Y,X)
      RETURN
   20 IF (Y.NE.0.)   GO TO 10
      CANG=PI
      RETURN
   30 IF (Y.GT.0.)   GO TO 40
      IF (Y.LT.0.)   GO TO 50
      CANG=0.
      RETURN
   40 CANG=HAFPI
      RETURN
   50 CANG=-HAFPI
      RETURN
      END
```

```
*DECK INDF
      SUBROUTINE INDF (HH,XX,EORH,F)
C THIS SUBROUTINE CALCULATES THE INDUCTION FIELDS FOR E SUB R AND
C H SUB PHI
C THESE INDUCTION FIELDS ARE FOR POSITIVE TIME FUNCTION
C EORH = 1. GIVES INDUCTION FIELD FOR E SUB R (TOWEL BAR)
C EORH = 0. GIVES INDUCTION FIELD FOR H SUB PHI (LOOP)
C FZ = INDUCTION FIELD FOR E SUB R
C FH = INDUCTION FIELD FOR H SUB PHI
      COMMON /ZOTA/ARRAY(15)
      COMMON /PIS/TWOPI,PI,HAFPI,QRTPI
      DOUBLE THETAD,SINTH,COSTH,R,CONS
      COMPLEX FZ,FH,F
      A =ARRAY(3) $C = ARRAY(9)*1.E+9 $WAVE = ARRAY(10)
      IF(XX.LE.0.)3,4
    3 PRINT 9500
9500  FORMAT (//* IN INDF, DISTANCE IS ZERO OR NEGATIVE, XX = *,F20.10)
      CALL EXIT
    4 THETAD=1.D0*XX/A
      SINTH=DSIN(THETAD)
      COSTH=DCOS(THETAD)
      R=A+HH
      CONS=A*R*SINTH**2
      D2=R*R+A*A-2.*A*R*COSTH
      IF(D2.GT.0.) GO TO 30
      DD=XX
      D2=XX*XX
      GO TO 40
   30 CONTINUE
      DD=SQRT(D2)
   40 CONTINUE
      D3=DD*D2
      D4=D2*D2
      FZR=-2.*COSTH/DD+3.*CONS/D3
      FZI=(WAVE*CONS+2.*COSTH/WAVE)/D2-3.*CONS/(D4*WAVE)
      FZ=CMPLX(FZR,FZI)
      FZ=FZ/CMPLX(0.,WAVE)
      FHR=R*SINTH*DD/D3
      FHI=WAVE*R*SINTH*DD/D2
      FH=CMPLX(FHI,-FHR)
      C1=2.E-7*TWOPI*WAVE*C
      FH=FH/C1
      IF(EORH)1,2
    1 F=FZ
      RETURN
    2 F=FH
      RETURN
      END
```

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

Printed by
United States Air Force
Hanscom AFB, Mass. 01731